MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Memo No. 499

September 1978

CAUSAL REASONING AND RATIONALIZATION IN ELECTRONICS

by

Johan de Kleer

Abstract:

This research attempts to formalize the type of causal arguments engineers employ to understand circuit behavior. A causal argument consists of a sequence of changes to circuit quantities (called events), each of which is caused by previous events. The set of events that an individual event can directly cause is largely an artifact of the point of view taken to analyze the circuit. A particular causal argument does not rule out other possibly conflicting causal arguments for the same circuit. If the actual behavior of the circuit is known or determined by measurements, the correct argument can be identified. The selected argument is a rationalization for the observed behavior since it explains but does not guarantee the observed behavior.

A causal analysis program QUAL has been implemented which determines the response of a circuit to changes in input signals. It operates with a simple four valued arithmetic of unknown, unchanging, increasing and decreasing. This program is used to illustrate the applicability of causal reasoning to circuit recognition, algebraic analysis, troubleshooting and design.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's research is provided in part by the National Science Foundation under Grant MCS77-04828.

Causal Explanations of Circuit Behavior

When an electrical engineer is asked to explain the operation of an electrical system he will often describe it in terms of a sequence of events each of which is "caused" by previous events. By throwing away most of the details of the system, he is able to extract a sequential description of the behavior of the system, characterizing its major features. This is sufficient for many purposes.

Sequential descriptions are ubiquitous in engineers' verbal and textbook explanations. Consider the Schmitt trigger (see Figure 1). The explanation reads as if a time flow has been imposed on it.

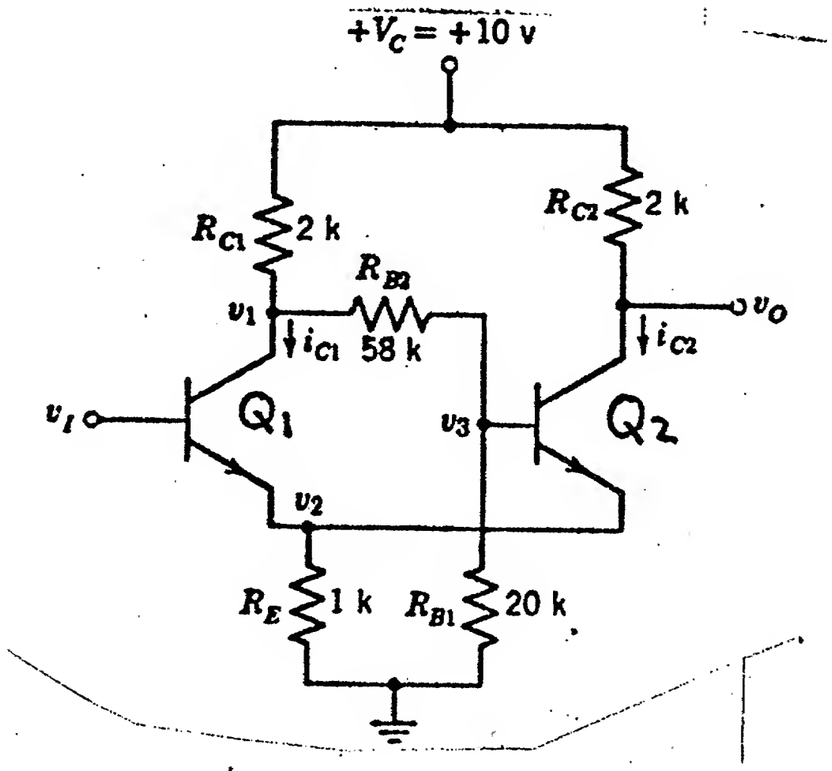

Figure 1 : The Schmitt Trigger

"... An increase in v_I augments the forward bias on the emitter junction of the first transistor, thereby causing an incremental increase in the collector current, i_{C1} of that transistor. Consequently both the collector-to-ground voltage v_1 of the first transistor, and the base-to-ground voltage of the second transistor v_3, decrease. The second transistor operates as an emitter follower which has an additional load resistor on the collector. Therefore, there is an decrease in the emitter-to-ground voltage v_2. This decrease in v_2 causes the forward bias at the emitter of the first transistor to increase even more than would occur as a consequence of the initial increase in v_I alone...."[Harris *et.al.* 66, p.68]

A goal of this research is to develop a clear understanding of the notion of causality as found in this argument.

Causal explanations describe how the behaviors of individual components contribute to the overall behavior of the circuit and are therefore useful in analysis, troubleshooting and design. Since a complete algebraic analysis of even simple circuits can be expensive, knowledge of how the individual components contribute to the circuit's composite behavior indicating which algebraic model should be used in the analysis, significantly improves efficiency [de Kleer & Sussman 78]. For example, an integrated circuit operational amplifier contains a large number of transistors, but few of them are situated on the main signal path. For many calculations the effect of these auxiliary transistors on the signal can be ignored or accounted for by much simpler transistor models. The causal explanation identifies which transistors are crucial to the behavior and which are not. Causal reasoning also plays a fundamental role in identifying the faults responsible for

symptomatic behavior and in localizing faults at a shallower level of detail before entering the more expensive deep analysis [Brown 76] [de Kleer 76]. Early designs can be checked to see whether they have any hope of achieving their desired behavior, and the sections which are critical to the desired behavior can be identified for special attention [McDermott 76].

Causality is an Artifact

The "causality" of an argument is an artifact of the level of detail used in the analysis that produced it. This can be demonstrated in the Schmitt trigger example by using a transistor model whose v_{BE} is fixed. Using this model v_3 still drops as a consequence of increasing i_{C1}, but v_2 now rises since v_I is rising and v_{BE} is fixed. Both of these effects cause v_0 to rise. This new argument predicts the same output signal, but the details of how this signal is achieved are completely different. The new argument does not identify the feedback, and predicts that v_2 will rise while the earlier argument predicts it will drop. This is an example of two different causal explanations for the external behavior.

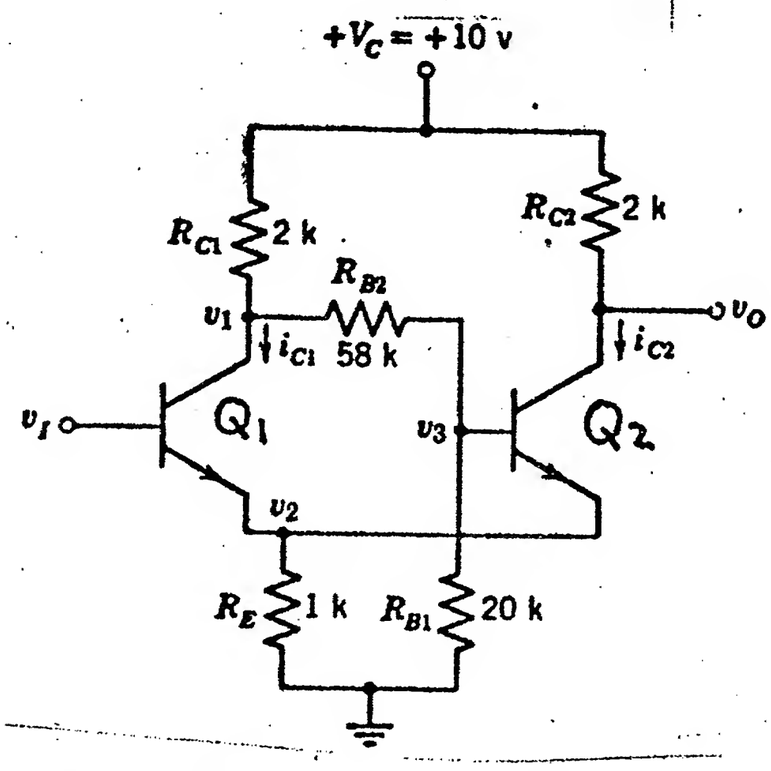

Figure 2 : The Schmitt Trigger

Since the component models utilized in these causal arguments are local, these arguments could all have been generated by a simple propagation of known signals: the signals are applied to the their adjacent device models which in turn predict other signals. Although most causal arguments can be generated by propagation, no such claim can be made about the validity of the converse. With a rule "A causes B," propagation will deduce B if A is valid, but will also deduce A if B is valid when there is no other plausible cause to account for it. The latter deduction is undesirable. For example, one usually thinks of increased v_{BE} causing increased i_C, but the inverse deduction of increased i_C causing increased v_{BE} is usually thought to be noncausal since something must have increased v_{BE}. However, the collector current cannot increase unless something supplied it with more current. This example further illustrates that "causality" is largely an artifact of the point of view taken to analyze the circuit.

The explanation for the Schmitt trigger made a number of unsubstantiated assumptions aside from the choice of transistor models. Why does the v_I increment appear across Q_1 instead of R_E? Why does the voltage v_1 drop since Q_2's turning off should raise it? Why is the current

contributed by Q_2's turning off more than the current taken by Q_1's turning on? There are many values for the parameters for which the circuit cannot function at all. Most explanations exhibit these kinds of inaccuracies. The arguments are only rationalizations of the observed behavior (observed by actual measurements or stated in the textbook). This does not detract from the usefulness of the explanations: no explanation ever accounts for every detail of the behavior. The usefulness of an explanation does not depend on how complete or correct it is, but whether the explanation is sufficient for the purposes it is applied to.

The Machinery for Causal Analysis

With the preceding discussions providing a philosophical starting point, this section develops a mechanistic model for causal reasoning. The purpose of the model is to explain how causal arguments can be discovered. A causal argument consists of a sequence of assertions about electrical quantities each of which hold as a consequence of previous assertions. For example, the causal argument "... An increase in v_I augments the forward bias on the emitter junction of the first transistor, thereby causing an incremental increase in the collector current,...", is a sequence of two assertions: v_I increases, i_{C1} increases. I call these assertions the *events* of a causal argument. The deduction of one event from another is determined by device models. In the above example the model for the first transistor is one in which increased emitter potential causes increased collector current. The device models are central to the theory since they utilize a description of the topological structure of the circuit to determine the mechanism underlying the behavior of the circuit. These models are the only part of the theory that refers to circuit topology; all further theory will utilize the mechanism fragments that the models produce.

The causal analysis machine is based on the presuppositions that the causal device rules are local and that the events of a causal argument are discovered in their causal sequence. By local I mean that the rules for a device (1) refer to a small number of circuit quantities, (2) refer to circuit quantities that are topologically adjacent to the device being modeled, and (3) that every device of the same type is modeled by the same rules, independent of topological context.

The causal analysis machine has three components. The modeling component specifies the behavior of the basic devices. The wiring component provides a way to describe circuits and circuit models. The execution component determines when device rules are to be applied. The wiring and execution components are almost completely determined by the presuppositions of localness and ordering. The modeling component will be discussed after the wiring and execution have been developed. Since the construction of device models is difficult, it is important to determine as much information as possible about their general structure before considering specific models.

The quantities of interest in the analysis of a circuit are represented by *cells*. Each voltage, current and device parameter has its own unique cell. A cell may contain one or more values. For example, i_{C1} may be represented by CELL-67 and contain the values 1 ma and 0 ma signifying that the collector current is 1 ma when Q1 is on and 0 ma when Q1 is off. Each cell is connected to the other cells by electrical laws. Whenever a cell receives a new value, the rules it participates in are examined to determine whether it is possible to deduce new values for

neighboring cells. Since a cell can participate in many rules, a queue of newly discovered values is maintained. This can introduce nondeterminism. If only one value can be deduced from each application of a rule, the queue will not grow and the assignments will be totally ordered.

The behavior of an electrical component is described by a device model which consists of an association list specifying the cells the model is connected to and rule prototypes referring to these cells. The rule prototypes specify how values in the cells are related. A transistor model utilized in the Schmitt trigger argument might be:

```
(((v (voltage e))                    association for emitter potential
  (ic (current c)))                  association for collector current

 (increasing v implies increasing ic))   rule prototype
```

Voltage and current refer to the appropriate cell of the specific transistor. The detailed structure of the rule prototypes will be discussed in the next section. When a new circuit is created, instances of the circuit models are created for each of the circuit's devices. An instance of a model is constructed by making a copy of the rule prototype and connecting it to the transistor's cells as indicated by the association list.

The wiring component provides a very general mechanism. SYN [de Kleer & Sussman 78] uses the same machinery to do synthesis of electronic circuits by propagation of constraints. In the case of propagation of constraints the rule prototypes are algebraic equations. The causal analysis described in this paper is implemented using this basic machine.

Causal flow analysis, which describes circuit behavior in terms of a sequence of events, is distinguished from other types of analysis by how it deals with time. Causal analysis assumes that the time of the basic machine can be identified with the sequential events of the causal argument, later events in the argument are discovered later in the analysis. Each event in a causal argument is an assignment of a value to a cell. This value depends on previous events in the argument, and must not be changed or improved upon after it has been placed there. Some of the consequences of this are that each cell is assigned a value only once and that each rule is used only unilaterally. A rule is used *unilaterally* if each of the cells it is connected to is used only as an output or as an input, but not both. If a rule uses the same cell as an input or output, it is used *bilaterally*. Analogously a rule which has the potential to be used bilaterally is referred to as bilateral rule.

Propagation of constraints violates most of these conditions when it introduces anonymous objects. Cell values which depend on anonymous objects change as the system solves for the anonymous objects. In order for propagation of constraints to solve for the anonymous objects, the rules must be expressed as bilateral constraints. The rules used in causal models, however, tend to be unilateral: transistor v_{BE} can cause i_C, but not vice versa. The conditions of causal flow analysis demand that every bilateral rule be used only unilaterally. For example, the causal resistor model is bilateral in i_R and v_R, but the rule must be used only unilaterally in a particular causal flow argument: for any resistor, i_R must be used to derive v_R, or vice versa, but not both.

An analysis by propagation of constraints that does not require the introduction of anonymous objects meets the criteria for a causal flow analysis. Such analyses are rare. A causal

flow analysis is permitted to make assumptions about the behavior of the circuit. Assumptions, like anonymous objects, are used to break impasses in the analysis. However, an assumption is not a kind of disguised anonymous object. The anonymous object is introduced in the hope that the ensuing propagations will be able to restrict the anonymous object's value. A propagation based on an assumption has a completely determined value and this value does not change if the assumption is validated or refuted. Assumptions provide a way of expressing partial information about the circuit's behavior. A value which depends on an anonymous object is unknown, but a value which depends on an assumption is known if the assumption is valid. The applicability of causal flow analysis depends on how easy it is to compute with these assumptions. Although it is easy to express assumptions represented as algebraic expressions, it is difficult to compute with them.

The machine can be controlled in two distinct ways. The queue of pending deductions can be reordered arbitrarily and the rules upon which it operates are arbitrary. These two techniques allow the machine to be controlled such that the implicit time order of its deductions is identified with the time imposed by a causal argument. The basic idea is that device models are forced to be *locally causal*. For example, the causal model for the transistor does not respond to changes in i_C. Deductions based on assumptions are inserted at the end of the queue.

The determination of output behavior is just one purpose of analysis. In this research, the main purpose of analysis is to determine how the individual components contribute to the circuit's overall behavior. Electrical engineering has developed an informal folklore of these contributions, and much of this folklore is related to causal flow analysis. For this reason, this research focuses on developing a theory of causal flow analysis for circuits. Such a theory, by itself, is useful for many purposes, but it also provides the foundation for a more sophisticated analysis of the teleology of the components.

Electrical Device Models

Although the constraint models for devices are agreed upon, no similar agreement exists for the causal models which are tacitly used in qualitative arguments. This section presents a sequence of different models for a few devices in order to explain the issues involved. A simple model will be proposed first, followed by more sophisticated models designed to correct the shortcomings of the first.

The causal explanation of how a circuit works is a qualitative description of the equilibrating process that ensues when signals are applied to the circuit. The behavior of the Schmitt trigger was described in this way. This will be called *incremental qualitative* (IQ) analysis. Since most circuits are designed to deal with changing input signals, it is not surprising that the main purpose of most circuits is achieved incrementally. For example, an amplifier must amplify changes in its input, digital circuits must switch their internal states as applied signals change, and power-supplies must provide constant current or voltage in the face of changing loads and power sources. For these kinds of circuits, the purpose of the quiescent behavior is to put the nonlinear devices into a desired region of incremental behavior. Since quiescent behavior plays a subsidiary role in the incremental analysis, this paper will concentrate on the latter.

Incremental qualitative arguments rarely need to refer to more than the sign of the

derivative which indicates whether the signal is increasing or decreasing. This requires an algebra of four values: "↑" signal is increasing, "0" signal is not changing, "↓" signal is decreasing, and "?" signal is unknown. The arithmetic of this algebra is very simple:

y: \ x:	↓	0	↑	?
↓	↓	↓	?	?
0	↓	0	↑	?
↑	?	↑	↑	?
?	?	?	?	?

Table 1 : x + y

Only addition and subtraction are important, and no other operations are ever used. Anonymous objects are never used in causal arguments and are thus unnecessary. These restrictions make the algebra subsystem of the machine trivial.

The plan for constructing the models is to start with the classical constraint models, and reformulate them preserving only the sign of the derivatives of the variables. Ohm's law has a particularly simple formulation.

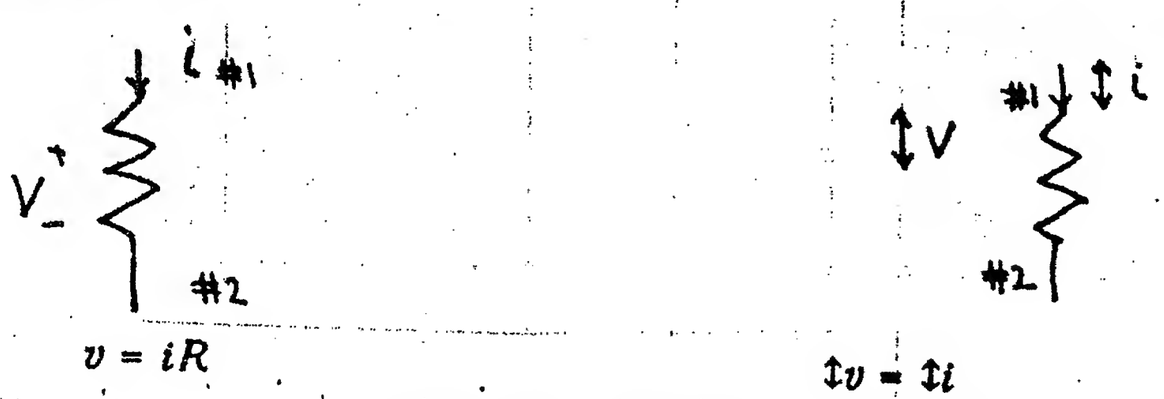

Figure 3 : Ohm's Law

↕x refers to the sign of the derivative of x. Models are described by an association list of variable bindings followed by a specification of the relationships which hold among these variables. The IQ model for Ohm's law is:

```
(((v (voltage #1 #2))
  (i (current #1)))
 (↕v <=> ↕i))
```

The #1 and #2 which appear in the association list refer to the two terminals of the resistor. Currents are defined to flow into devices away from nodes. Kirchoff's Current Law (KCL) applies to components so that the current in #1 is equal and opposite to the current through terminal #2. The rule prototype of the model specifies that the derivative of the current must be the of the same sign as the derivative of the voltage. Since the resistor has no preferred causal flow direction this rule must be bilateral. This action is specified by the "<=>" operator.

The ideal diode conducts zero current when the voltage across it is below a certain threshold and conducts an arbitrary amount of current at that threshold. This behavior is usually modeled by the two states on and off:

```
(((v (voltage #1 #2))
  (i (current #1)))
 (choice (on  (0 => ↕v))
         (off (0 => ↕i))))
```

The choice construct specifies the rule prototypes that apply for each of the regions of operation of the device. The "=>" operator is like <=> except that it specifies an assignment in one direction only. In the off state, the current through the diode is zero as well as all of its derivatives. The above model, however, only indicates that the current is unchanging (i.e. the first derivative is zero). A particularly simple model for a transistor has an ideal diode as its emitter junction and a controlled current source at its collector:

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on (0 => ↕v) (↕ib => ↕ic) (↕ib -=> ↕ie))
         (off (0 => ↕ib) (0 => ↕ic) (0 => ↕ie))
         (sat (0 => ↕v) (0 => ↕ic))))
```

Note that => and <=> always refer to derivatives. The "-=>" operator behaves like => except that it inverts the sign of the assigned quantity.

Analysis of a DTL-Inverter

In order to analyze a circuit containing devices which have different states, the various composite circuit states must be considered. Sometimes the applied signal can force a unique state choice, and sometimes a number of possible circuit states have to be explored simultaneously. Transistor and diode models assert values which are dependent only upon the state they are in. A *state-value* assertion for a transistor in the off state is (0 => ↕ib). State-value assertions can be invoked without propagations, but in order to prevent a proliferation of circuit states, the state-values of a model are only used if a signal is detected near the device. In this way new circuit states will only be considered when necessitated by the propagation. A signal can also cause the circuit to change state. The rules for such state transitions will be discussed later. Adopting the convention that the state-values are listed separately before the other rules, the transistor model is described as:

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ((0 => ↕v) (↕ib => ↕ic))
             (↕ib -=> ↕ie))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie)))
         (sat ((0 => ↕v) (0 => ↕ic)))))
```

These are enough device models to analyze the simplified DTL (Diode-Transistor-Logic)

inverter which is constructed from transistors, diodes and resistors:

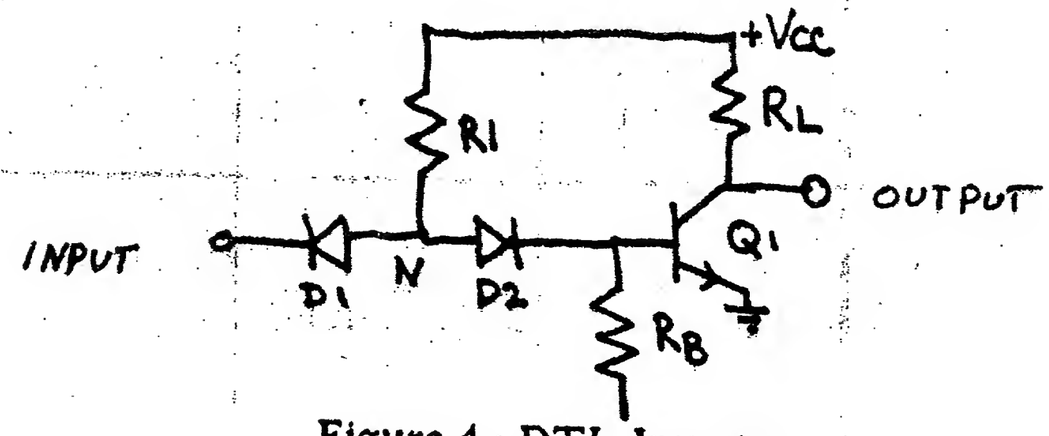

Figure 4 : DTL-Inverter

When a voltage signal is applied to the input, nothing happens since the diode model only operates on current through the diode, or voltage across it. Since the simple diode model only has outputs, the analysis must make an arbitrary choice as to whether D1 is on or off. If D1 is off, the current through it is zero and propagation halts indicating that the rest of the circuit values remain unchanged. If D1 is on, an increase in input voltage results in an increase at N. A similar analysis applies to D2. If D2 is off, the remaining circuit values are unchanged. If D2 is on, the voltage at the base of Q1 is rising which is only possible if Q is off. If Q1 is on, the model says that its base-emitter voltage cannot vary. The analysis of the DTL-inverter fails.

The DTL analysis failed to explain how the inputs to the circuit affect its output. The ideal diode model for Q1 produces a contradiction when Q1 is on. Even ignoring the contradiction, the models for D1 and D2 do not say anything about the current flowing through them. Therefore no signal would appear at i_B or the output node. One possible solution is to include the exponential diode effect for Q1.

```
(choice (on (↕v => ↕ib) (↕v => ↕ic) (↕v -=> ↕ie)))
```

This evades the contradiction. Unfortunately, if this exponential diode model is used for D1, the analysis can no longer determine whether the voltage at N drops. D1 and D2 could have their polarities reversed without affecting the analysis:

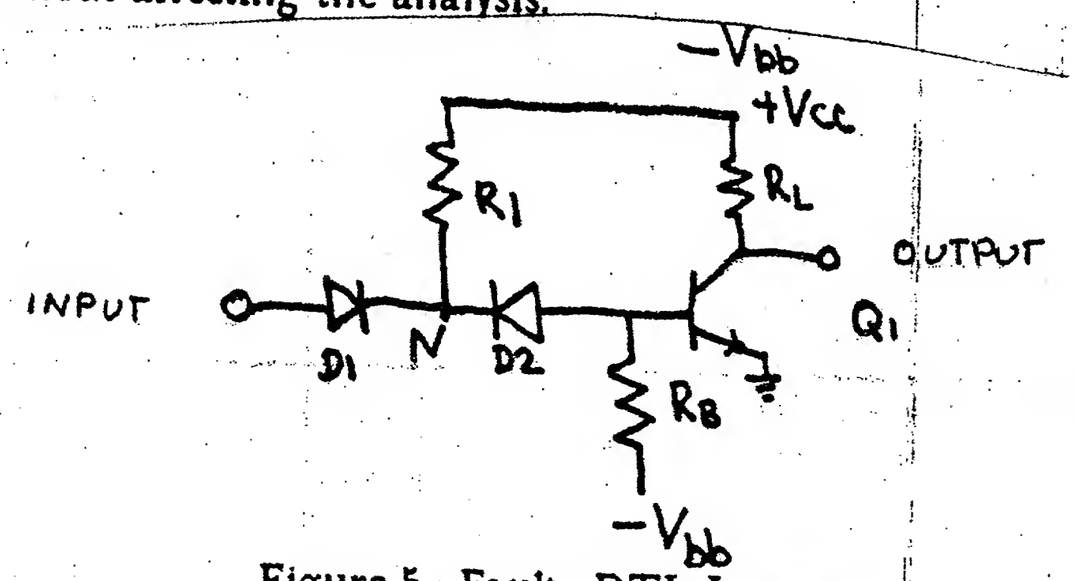

Figure 5 : Faulty DTL-Inverter

In the correct circuit the current through D1 decreases as the input signal rises. In the faulty circuit this current increases. Since no external voltage is discovered across D1, the exponential diode model cannot be utilized to determine the direction of current flow.

Consider a causal argument a person might give for the inverter's operation: "As the input signal rises, N rises and the current through D1 drops. As N rises, D2 turns on harder, increasing the current through it and raising the base of Q1. Q1 turns on harder and pulls down

the output." Note that each device appears only once and its model is often invoked upon insufficient evidence. For example, D1 can only communicate the signal to N if the voltage at N is higher if D1 is removed. The current through D1 decreases only if N does not rise faster than the input. The model employed to describe D1 makes the presupposition that this is case. Stated differently, the diode model always makes the presupposition that the first signal detected near the diode invokes the model as if this signal dominates all of the other quantities the model references. In this simple circuit these presuppositions can be trivially verified, but there is no way the diode model, which only has access to local information, can determine this.

This is the the beginnings of the notion of a causal argument. To reiterate, a causal argument consists of a sequence of events, each event describing how the behavior of a node or device is influenced by earlier events, with the presupposition that the discovered trigger signal is the dominant input to that node or device. Assuming that the ordering of the events within the execution component can be identified with the sequence of the causal argument, the *causal presupposition* can be incorporated into the models. The diode model now becomes:

```
(((v (voltage #1 #2))
  (v1 (voltage-to-reference #1))
  (v2 (voltage-to-reference #2))
  (i (current #1)))
 (choice (on ()
             (↕v (=> ↕i))
             (↕v1 (C=> ↕v2) (C=> ↕i) (C=> ↕v))
             (↕v2 (C=> ↕v1) (C-=> ↕i) (C-=> ↕v)))
         (off ((0 => ↕i)))))
```

The choice construct has been slightly modified. The first expression of a choice lists the state-values, and the remaining rule prototypes are grouped together according to input variable. The "C=>" operator acts like =>, except that it acts only on nonzero values. In order to include the consequences of the causal presupposition explicitly, models refer to voltages at their terminals as well as voltages across their terminals. The causal presupposition assumes all values are zero, so it is never necessary to propagate zero values. In fact, a zero input should never be considered a dominant input, even if it is found first. In those cases where the zero value would have participated in a contradiction, the value it would have contradicted with must be nonzero and that value will have propagated causing a contradiction at a slightly different place.

The model for a transistor now becomes:

```
(((v (voltage b e))
  (ve (voltage-to-reference e))
  (vb (voltage-to-reference b))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ()
             (↕v (=> ↕ic) (-=> ↕ie) (=> ↕ib))
             (↕vb (C=> ↕ve) (C=> ↕v) (C=> ↕ib) (C-=> ↕ie) (C=> ↕ic))
             (↕ve (C=> ↕vb) (C-=> ↕v) (C-=> ↕ib) (C=> ↕ie) (C-=> ↕ic)))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie)))
         (sat ((0 => ↕ic)))))
```

If the transistor is directly connected to the local reference, ve and vb are not utilized. This is the case with the DTL-inverter.

Employing these models the DTL analysis succeeds. The following is the causal argument that QUAL finds for the output behavior. The format of this explanation is a causally-ordered sequence of events described by cell-value pairs, each of which is followed by a one line explanation of the model rule that deduced it. Since events can have multiple antecedents and consequents, only simple causal arguments can be expressed with a totally ordered linear list. When an event has multiple consequents or antecedents this fact will be indicated in the causal argument and the argument for that value will be included in a judicious place in the event sequence. In general, there are many events caused by the inputs which do not affect circuit outputs. These will not be included in the causal arguments.

```
Starting with input:
(VOLTAGE INPUT GROUND) = ↑
   Premise.
(VOLTAGE N GROUND) = ↑
   V2 C=> V1 for D1
(VOLTAGE B GROUND) = ↑
   V1 C=> V2 for D2
(CURRENT C Q1) = ↑
   V => IC for Q1
(CURRENT #2 RL) = ↓
   KCL for node OUTPUT
(CURRENT #1 RL) = ↑
   KCL for device RL
(VOLTAGE OUTPUT VCC) = ↓
   = V I for RL
Also given that:
(VOLTAGE GROUND VCC) = 0
   POSITIVE-SUPPLY
```

The combination of events (VOLTAGE VCC GROUND) (VOLTAGE VCC OUTPUT) cause:
(VOLTAGE OUTPUT GROUND) = ↓
  KVL applied to nodes GROUND VCC OUTPUT

The deductions the models make depend upon the order in which the propagator discovers new values. Suppose a rising voltage is applied to a transistor. If the increase is applied to the base, the emitter must follow and the collector current increases. If the increase is applied to the emitter, the base will rise and the collector current decreases. Taking into account only the voltages at the base and emitter, the two examples are identical. The collector current is determined by which voltage the propagator found first. The causal presupposition says that the collector current is determined by which voltage caused the other.

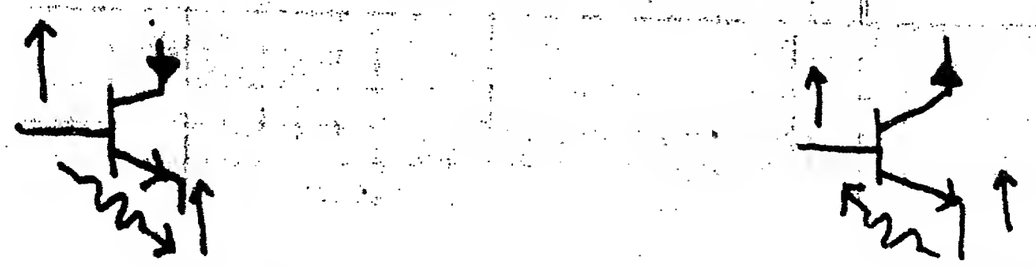

Figure 6 : Causality at the Emitter Junction

The causal presupposition can be violated, and the propagator must detect these violations. Whenever a model makes a deduction based on the presupposition it should explicitly mention which values are assumed to be zero with respect to triggering quantity. If this assumption is ever violated, the propagator should retract the original deduction. Causal presuppositions can also make subsequent teleological reasoning more difficult. The sole purpose of a circuit fragment may be to ensure the nondominance of a quantity. If a causal presupposition is made that this quantity is nondominant, the purpose of the circuit fragment cannot be determined. To avoid this difficulty, the propagator should try to substantiate all of its causal presuppositions after the analysis is completed.

Connection Heuristics

The rules of the device models are of two different types: rules which involve assumptions that do not necessarily hold, and basic rules which involve no assumptions and are universally valid. The $\updownarrow v_{BE} \Rightarrow \updownarrow i_C$ transistor rule makes no assumptions and is thus a basic rule. An example of an heuristic rule is $\updownarrow v_B \Rightarrow \updownarrow i_C$ which assumes that the v_B input is dominant. In order to reason about and possibly retract these assumptions, the assumptions themselves have to be explicitly recorded.

The heuristic rule $\updownarrow v_B \Rightarrow \updownarrow i_C$ makes an assumption about the behavior of the circuit around the transistor and not about the transistor itself. If this heuristic voltage rule is consistently applied to all the device models, every basic voltage rule must be expanded into two heuristic voltage rules. These voltage rules specify how the individual device models are connected together, and therefore a special *KVL connection heuristic* is introduced to replace them. The KVL-heuristic is implemented as a procedure which is triggered whenever a nonzero voltage is discovered at a node. It looks for device models with voltage inputs that refer to this node and triggers them. For

example, when the KVL-heuristic discovers a voltage at the base of a transistor, it triggers the model on its base-emitter voltage. In doing so, the rule makes the assumption that the emitter voltage is negligible with respect to the base voltage. The assumption that the base voltage is the dominant input to Q is recorded as $[Q\ v_B]$. Under this assumption a rising base voltage will thus cause a rising collector current.

If the voltage at the emitter is discovered to be rising independently, the KVL-heuristic determines that the collector current is falling under the assumption $[Q\ v_E]$.

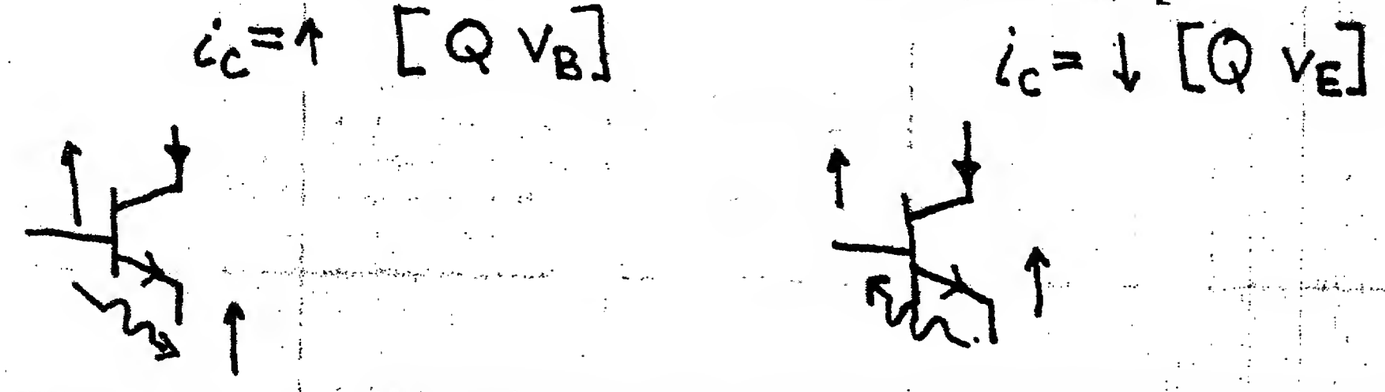

Figure 7 : v_B and v_E Assumptions

Since the two contradicting values for i_C hold under different assumptions, the only effect of the contradiction is to record that at least one of the assumptions is invalid. The introduction of explicit assumptions has freed the analysis process from the nondeterminism introduced by the queue; no matter when v_B is discovered, it will propagate to i_C since that propagation step involves a new independent assumption different from any other assumptions that were made about that transistor.

KVL is inherently a constraint law. One possible causal implementation of this constraint attempts all possible consistent assignments of values to the individual branch voltages. If the quiescent current flow directions are known, the situation is improved but the strategy generates far too many assumptions to be useful. Instead, the KVL-heuristic assigns a value only to the outermost branch voltage, other rules being expected to propagate this voltage to the individual branches. The input voltage of the Schmitt trigger appears across the input transistor and emitter resistor. Since the transistor is connected to the input voltage, it is the device that receives the input voltage rather than the resistor which is connected to the neutral reference. The voltage across the resistor must be calculated by the other rules.

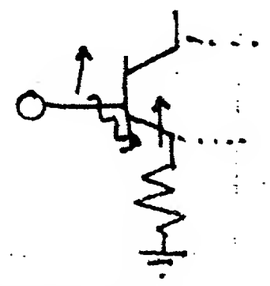

Figure 8 : Schmitt Trigger Input

From an equilibrium point of view this KVL-heuristic is false, but it captures the kind of causality manifested in the Schmitt trigger explanation. The KVL-heuristic also makes the presupposition that all interesting voltages eventually propagate to a voltage with respect to a common reference. This presupposition is false in analog multipliers and other heuristics have to be developed to deal with such circuits. These can be analyzed if more references are introduced, but this results in excessive redundant arguments as well as requiring *a priori* knowledge of circuit behavior.

Associated with each propagated value is an *environment* descriptor which indicates the circuit state it applies to and the assumptions under which it is valid. If incompatible environments are kept separate, different environments can be explored simultaneously. In this way those areas of circuit behavior which are common among environments can be shared. Two environments are incompatible if one environment contains an assumption or state choice on a device and the other environment contains a different assumption or state choice on this same device. Thus any environment which contains $[Q\ v_B]$ is incompatible with any environment which contains $[Q\ v_E]$.

The rules of the device models come from the algebraic models used in electrical engineering and from the rough qualitative models observed in engineers' arguments. Since there is a diversity of algebraic and qualitative models, there is also a variety of incremental qualitative models. The standard algebraic incremental and quiescent models (Hybrid-π and Ebers-Moll) for a transistor employ a dependent current source to describe the collector current. A current source only constrains current and not voltage, therefore the IQ transistor model describes the collector current as a causal output and ignores the collector voltage. The causal action of the emitter junction is more complex, and the IQ model is based on the observed arguments engineers use. A simple model has $0 \Rightarrow \updownarrow v_{BE}$ and i_B as a causal input. In most situations the diode behavior is necessary to explain i_C (although $0 \Rightarrow \updownarrow i_B$, the infinite-beta model holds more generally). Although the exponential diode equation does not distinguish between voltage and current, the diode action is almost invariably described as a voltage causing a current, as seen in the fact that the diode equation is always written as an exponential. Mathematically, a logarithmic equation is just as accurate. Therefore the basic IQ transistor model treats v_{BE} as a causal input and i_B (if beta is finite) and i_C as outputs. The only rule an IQ model must obey is that it assert all the voltages and currents associated with the device, because a device model cannot trigger on its own outputs (or any consequence thereof).

An examination of electronics textbooks shows the dominance of voltage as a causal quantity. For example, voltage is explained as a force and current as the stuff moved by this force. The various IQ models and heuristics follow this convention. Mathematically there is no reason to distinguish between voltage and current and there are some circuits which are better understood in terms of currents, but they are relatively rare and will not be discussed here.

The IQ model for a transistor is represented as follows:

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ()
             (↕v (=> ↕ic) (-=> ↕ie) (=> ↕ib)))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie)))
         (sat ((0 => ↕ic)))))
```

When a voltage is discovered at a node, any causal input voltage which refers to that node (and some other) is assumed to also receive this voltage value.

The exponential diode model used in the DTL-inverter analysis has a rule which states that the voltage on the anode follows the voltage on the cathode. This rule makes assumptions about the behavior of the rest of the circuit. In particular, the rule assumes that the diode is not connected to a negative resistance. There are many other situations in which it is useful to make this kind of an assumption. The emitter junction of a transistor behaves as an exponential diode in that the voltage on the emitter usually follows the voltage on the base. Increased collector current usually pulls down the voltage on the collector node.

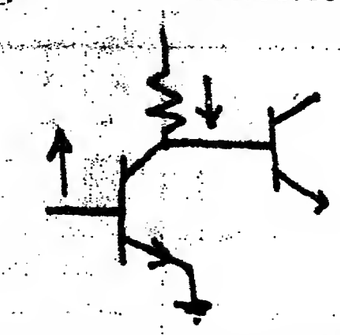

Figure 9 : Collector Current Pulling down Node Voltage

If the current through a resistor is caused to drop, the voltage at its positive terminal usually drops as well. In all these situations, current flow into a node affects voltage at the node. One way to make these kinds of deductions possible is to follow the example of the simple diode model and add heuristic rules to every model with causal current outputs.

This unnecessarily complicates the device models and requires even the basic models to make assumptions. This heuristic is really a statement about the behavior of the rest of the circuit, and not about the particular device causing the changing current. For these reasons a separate node model is used which models the behavior of nodes. The node model cannot be represented conveniently in the format used to express device models. The nearest approximation to the model for a n-terminal node is:

```
(((v (voltage-to-reference node))
  (i1 (current <terminal1>))
  ...
  (in (current <terminaln>)))
 (↕i1 (?-=> ↕v))
 ...
 (↕in (?-=> ↕v)))
```

The "?-=>" operator records the assumption that the current triggering the rule is dominant by [<node> <terminal>].

The procedure which implements the *KCL-heuristic* is more sophisticated. If the node voltage is unknown, and some of the currents into the node are known, then the voltage at the node rises if the sum of the currents ignoring KCL on the node is positive, and drops if the sum is negative. This assumption is recorded as [<node> <terminal1> ... <terminaln>]. The KCL-heuristic must be applied to every environment individually since a voltage known in one environment can be unknown in another. Since the KCL-heuristic makes such a major assumption about circuit behavior, and since it can be more judiciously applied if more currents and voltages are known, it is run after all possible propagations have been made in the environment onto which it will assert the new node voltage.

From the point of view of network theory, the KCL-heuristics make the assumption that the terminals which are causing current flow into the node can be modeled as the terminal of a current source, and that the remaining terminals can be modeled as the terminal of a positive resistance:

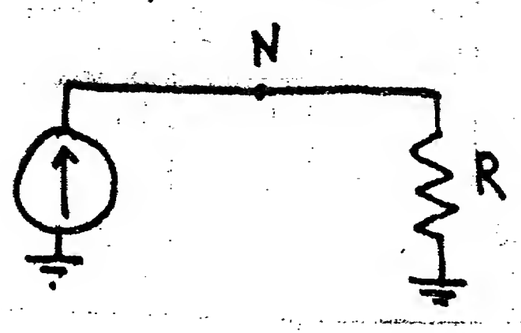

Figure 10 : KCL-Heuristic Network Assumption

The KCL-heuristic assumption can be violated. This is especially true in circuits with feedback. The KCL-heuristic can also be redundant in that the voltage at the node is either irrelevant or can be deduced in some other way.

In summary, the basic IQ machine employs three kinds of rules: model rules, KCL and KVL, and KCL- and KVL-heuristics. The rules of the device models are locally causal and do not make assumptions. KVL and KCL apply when all but one of a collection of currents or voltages is unknown. They also make no assumptions. Finally, the KCL- and KVL-heuristics allow the analysis to connect together the behavior of the local device models. Since KCL and KVL are inherently constraint-like, these two heuristics introduce an artificial equilibration time by making their assumptions explicit.

One purpose of the assumptions is to free the IQ analysis from the nondeterminacy of the queue of the basic propagation machine. If every possibly invalid event makes an explicit assumption, the order in which the events are found will have no effect on the ensuing contradictions. A second equally important purpose of assumptions is to identify the cause of a contradiction. The IQ rules may make far more assumptions than necessary. For example, no assumptions are logically necessary in the causal analysis of the DTL-inverter because all the IQ rules are completely local. In the case of the KCL-heuristic at node N, it was unable to tell that there was not a feedforward path to the top of R1. A rudimentary topological analysis could have determined this, but the KCL-heuristic cannot do any topological analysis and therefore must be prepared for the worst. There is however, a simple strategy to remove many of the assumptions. At the conclusion of the analysis all the possible causes have been investigated, and therefore any assumption that does not immediately lead to multiple values must hold. By this strategy, all the assumptions made in the DTL analysis are verified.

The IQ diode model is now very simple:

```
(((v (voltage #1 #2))
  (i (current #1)))
 (choice (on ()
             (↕v (=> ↕i)))
         (off ((0 => ↕i)))))
```

The following is the causal argument for the DTL-inverter utilizing the new models. Each event is followed by the list of assumptions (the environment) made by the causal argument .

```
Starting with input:
(VOLTAGE INPUT GROUND) = ↑  <>
   Premise.
(VOLTAGE N INPUT) = ↓  <[D1 V2] (D1 ON)>
   KVL-heuristic [D1 V2]
(CURRENT #1 D1) = ↓  <[D1 V2] (D1 ON)>
   V => I for D1
(VOLTAGE N GROUND) = ↑  <[N D1] [D1 V2] (D1 ON)>
   KCL-heuristic [N D1]
(VOLTAGE B N) = ↓  <[D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KVL-heuristic [D2 V1]
(CURRENT #1 D2) = ↑  <[D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   V => I for D2
(CURRENT #2 D2) = ↓  <[D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL for device D2
(VOLTAGE B GROUND) = ↑  <[B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL-heuristic [B D2]
(CURRENT C Q1) = ↑  <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   V => IC for Q1
(CURRENT #2 RL) = ↓  <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL for node OUTPUT
(CURRENT #1 RL) = ↑  <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL for device RL
(VOLTAGE OUTPUT VCC) = ↓  <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   = V I for RL
Also given that:
(VOLTAGE GROUND VCC) = 0  <>
   POSITIVE-SUPPLY
The combination of events (VOLTAGE VCC GROUND) (VOLTAGE VCC OUTPUT) cause:
(VOLTAGE OUTPUT GROUND) = ↓
      <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KVL applied to nodes GROUND VCC OUTPUT
```

Since no conflicting multiple values are found, the assumptions <[B D2] [N D1] [D2 V1] [D1 V2]> are verified.

The analysis of the emitter-coupled pair further illustrates the use of the connection heuristics.

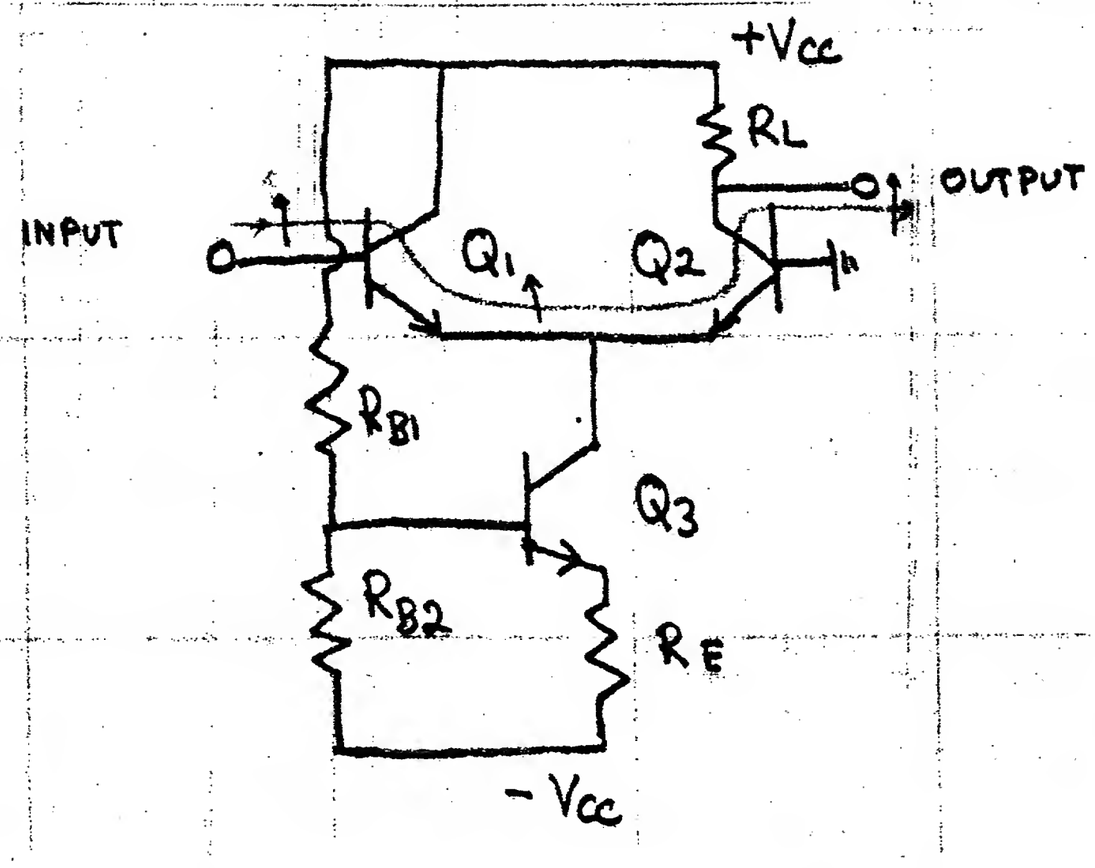

Figure 11 : Emitter-Coupled Pair

```
Starting with input:
(VOLTAGE INPUT GROUND) = ↑  <>
   Premise.
(VOLTAGE C3 INPUT) = ↓  <[Q1 VB] (Q1 ON)>
   KVL-heuristic [Q1 VB]
(CURRENT E Q1) = ↓  <[Q1 VB] (Q1 ON)>
   V => IE for Q1
(VOLTAGE C3 GROUND) = ↑  <[C3 Q1] [Q1 VB] (Q1 ON)>
   KCL-heuristic [C3 Q1]
(CURRENT C Q2) = ↓  <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   V => IC for Q2
(CURRENT #2 RL) = ↑  <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   KCL for node OUTPUT
(CURRENT #1 RL) = ↓  <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   KCL for device RL
(VOLTAGE OUTPUT +VCC) = ↑  <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   = V I for RL
Also given that:
(VOLTAGE +VCC GROUND) = 0  <>
   SUPPLY1
The combination of events (VOLTAGE GROUND +VCC) (VOLTAGE OUTPUT +VCC) cause:
(VOLTAGE OUTPUT GROUND) = ↑  <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   KVL applied to nodes OUTPUT +VCC GROUND
```

The input voltage is applied to the emitter junction of Q1 causing an increase in its emitter current thereby pulling up the voltage on the emitter. This reduces the base-emitter voltage of Q2 causing its collector current to decrease. Since the current flows through RL, the output voltage drops. This causal argument makes the assumption that the increased input voltage appears across Q1,

and that the emitter of Q2 and the collector of Q3 behave as a positive resistance. The first assumption is a result of applying the KVL-heuristic and the second assumption is the result of applying the KCL-heuristic. Because the KCL-heuristic is not applied to the output node, Q2's collector current can be used to deduce the output voltage without making an assumption. Since no signal is ever detected around Q3, all the circuit quantities around Q3, RB1 and RB2 are presumed to be zero. A transistor with no incremental collector current must be fulfilling some quiescent role. In this example Q3 is functioning as a current source.

The complexity of a causal argument depends on the device models used in the analysis. The simpler ideal diode model is sufficient to analyze most circuits. For example, the ideal diode model can explain the DTL-inverter's output behavior. Beta is not easily controlled in transistor fabrication, and so few circuits depend critically on it. For these circuits beta can usually be presumed to be infinite with the base current always zero. Some circuits, notably TTL gates, depend on a fourth region of operation of the transistor. The inclusion of this state unnecessarily complicates the analysis of other circuits, most of which do not depend on it. Since the incorrect choice of oversimplified models usually results in a failure to explain the behavior or an unretractable contradiction, the analysis can always start with simpler models and introduce the more sophisticated models if problems are encountered.

Recognition and Rationalization

The propagator can now generate a possible explanation for how the DTL-inverter works. This explanation is a rationalization, carrying no guarantee that the inverter functions. The DTL-inverter has 12 possible states, and the analysis reveals that if the circuit is an inverter, inversion must take place in the one state where all devices are on. This is a kind of recognition, answering the question "Could x perform function y?" [de Kleer 77]. Moreover, it gives a causal explanation of how that function could be achieved.

Since circuit can have state, the response of a circuit to a signal can be a transition from one state to another. Individual devices change state when the signals applied to them change, and thus incremental analysis can determine possible state transitions and their causes. Although the possible states cannot be verified without doing a quiescent analysis, incremental analysis can determine all the possible state transitions the circuit might follow in response to an input signal.

An example of a transition rule for the npn transistor is: if the v_{BE} is increasing and the transistor is off, it may eventually turn on. Rules of this kind fit neatly into the device models:

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ()
             (↕v (=> ↕ic) (-=> ↕ie) (=> ↕ib)
                 (if ↑ (S-> sat))
                 (if ↓ (S-> off))))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie))
              (↕v (if ↑ (S-> on))))
         (sat ((0 => ↕ic))
              (↕v (if ↓ (S-> on))))))
```

The expression `(if ↓ (S-> off))` indicates that a possible transition to the off state may occur if the signal is falling.

The model for a diode is much simpler:

```
(((v (voltage #1 #2))
  (i (current #1)))
 (choice (on ()
             (↕v (=> ↕i) (if ↓ (S-> off))))
         (off ((0 => ↕i))
              (↕v (if ↑ (S-> on))))))
```

When these transition models are used in the incremental analysis of the DTL-inverter, four possible state changes are found:

TRANSITION-RULE-4

<(D1 . ON)> --> <(D1 . OFF)>

Cause: (VOLTAGE INPUT GROUND) = ↑

If the input diode is on, a rising input voltage may eventually cause it to turn off.

TRANSITION-RULE-3

<(D2 . OFF) (D1 . ON)> --> <(D2 . ON) (D1 . ON)>

Cause: (VOLTAGE N1 GROUND) = ↑

If the input diode is on, its anode must be rising with the input signal. Thus, if the drop diode is off, it may eventually turn on.

TRANSITION-RULE-2

<(Q1 . OFF) (D2 . ON) (D1 . ON)> --> <(Q1 . ON) (D2 . ON) (D1 . ON)>

Cause: (VOLTAGE BASE GROUND) = ↑

If both diodes are on, the rising input is communicated to the base of the transistor and if it is off it may eventually turn on.

```
TRANSITION-RULE-1
<(Q1 . ON) (D2 . ON) (D1 . ON)> --> <(Q1 . SAT) (D2 . ON) (D1 . ON)>
Cause: (VOLTAGE BASE GROUND) = ↑
```

If both diodes are on, the rising input is communicated to the base of the transistor and if it is on, it may eventually saturate.

Applying these three transition rules to the 12 possible states results in 11 possible transitions between states. The circuit's states are described by (D1's state, D2's state, Q1's state):

```
TRANSITION-11: (ON OFF SAT) -->> (ON ON SAT) [TRANSITION-RULE-3]
TRANSITION-10: (ON OFF SAT) -->> (OFF OFF SAT) [TRANSITION-RULE-4]
TRANSITION-9: (ON OFF OFF) -->> (ON ON OFF) [TRANSITION-RULE-3]
TRANSITION-8: (ON OFF OFF) -->> (OFF OFF OFF) [TRANSITION-RULE-4]
TRANSITION-7: (ON OFF ON) -->> (ON ON ON) [TRANSITION-RULE-3]
TRANSITION-6: (ON OFF ON) -->> (OFF OFF ON) [TRANSITION-RULE-4]
TRANSITION-5: (ON ON SAT) -->> (OFF ON SAT) [TRANSITION-RULE-4]
TRANSITION-4: (ON ON OFF) -->> (ON ON ON) [TRANSITION-RULE-1]
TRANSITION-3: (ON ON OFF) -->> (OFF ON OFF) [TRANSITION-RULE-4]
TRANSITION-2: (ON ON ON) -->> (ON ON SAT) [TRANSITION-RULE-2]
TRANSITION-1: (ON ON ON) -->> (OFF ON ON) [TRANSITION-RULE-4]
```

These state transitions correspond to the following state diagram:

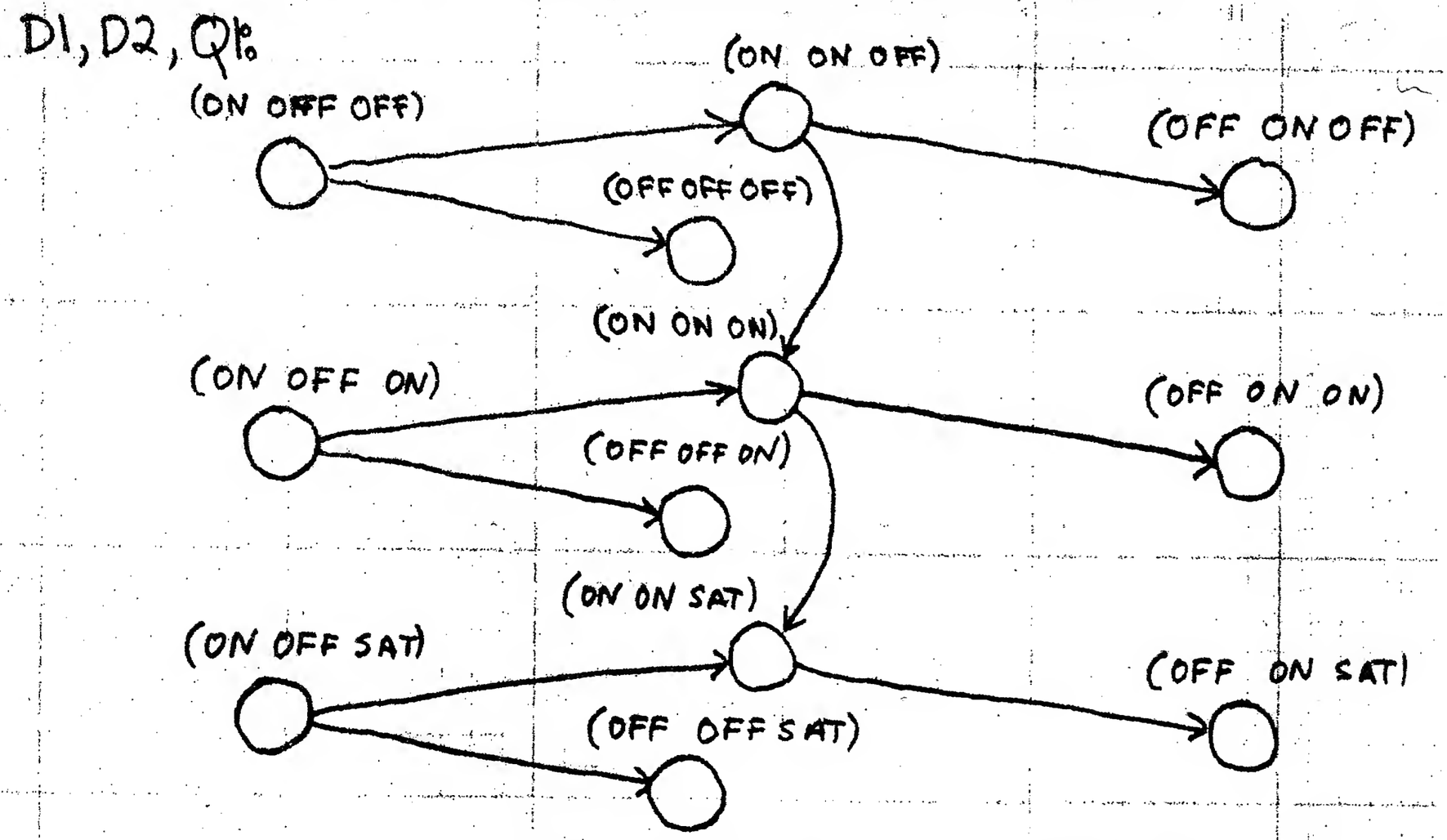

Figure 12 : State Diagram for DTL-inverter

Any state in which D1 is off has no outgoing transitions, because no signal can be communicated to the rest of the circuit when the input diode is off. The analysis cannot determine whether D1 turns off first or whether D2 turns on first. This is reflected in TRANSITION-RULE-4

and TRANSITION-RULE-3. A quiescent analysis could determine that (OFF OFF ?) was an impossible state. If vcc is more than two diode drops above ground, current must be flowing through R1 and one of D1 or D2 must be on. Further quiescent analysis could eliminate more of these states, but most of them can be eliminated by applying some simple heuristics to the state diagram.

In order to exhibit useful behavior, a circuit must respond to input signals. This simple *non-autism* rule substantially reduces the state diagram. For example, the state (OFF OFF OFF) can be eliminated because it can only be preceded by state (ON OFF OFF), and the output is zero in both states. The same argument applies to state (OFF OFF SAT). State (OFF OFF ON) is eliminated since it and the preceding state always have a rising signal. The new state diagram is:

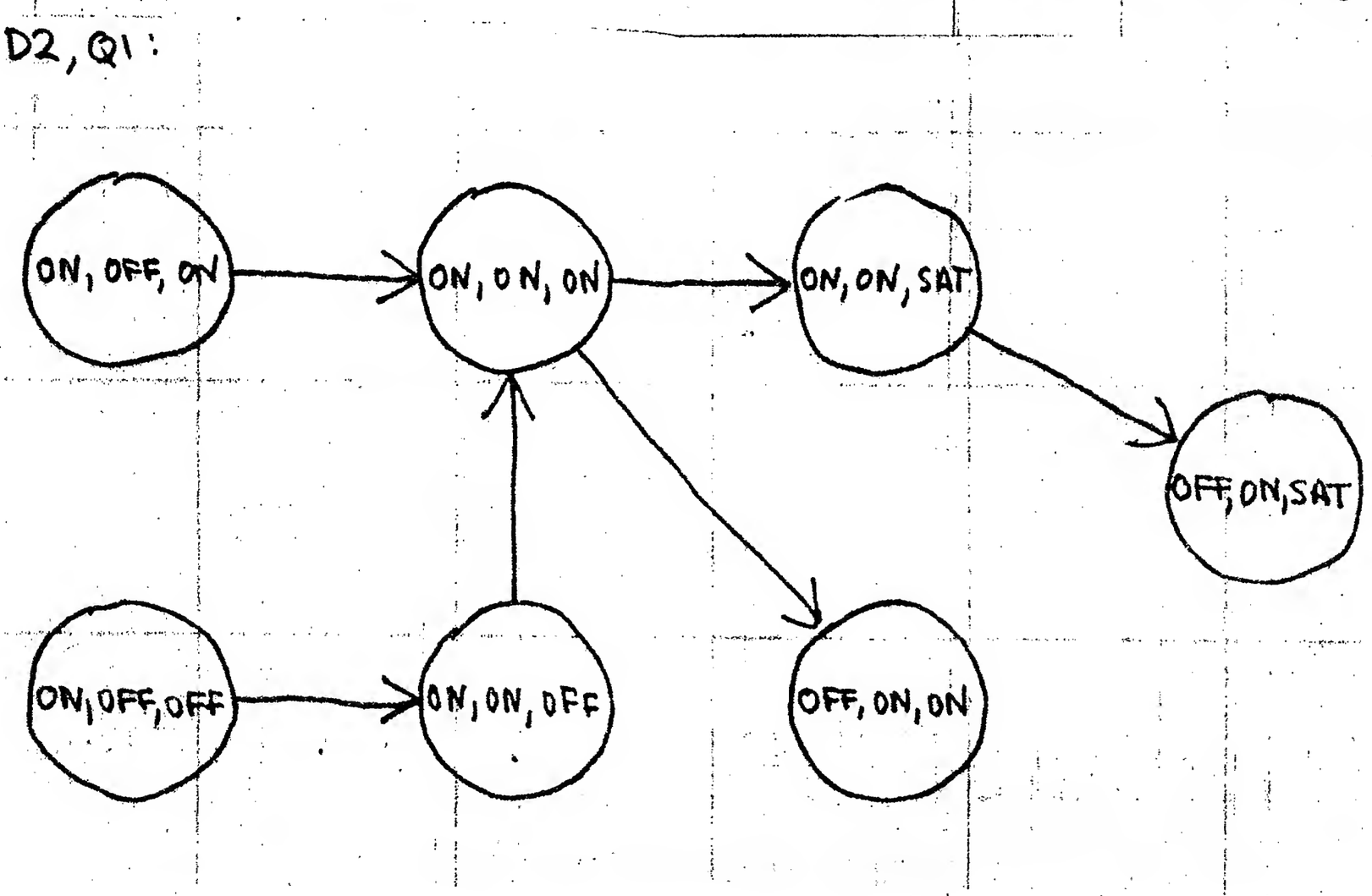

Figure 13 : State Diagram for DTL-Inverter after Simplification

The (ON OFF ON) state is impossible and could be ruled out by a simple quiescent analysis. The (ON OFF OFF) and (OFF ON ON) states can only be ruled out by knowing how the DTL-inverter is intended to operate.

Rules of this kind are insufficient to deal with all behaviors. Fortunately, this is not the goal of this endeavor. Determining a circuit's function solely from its schematic is, in general impossible, and rarely interesting. Instead, the unsimplified state diagram can be used to determine whether the circuit could perform a specified function. The DTL circuit is supposed to be an inverter. Applying the restriction that the circuit inverts to the original state-diagram (figure 12) also results in the simplified state-diagram (figure 13). The circuit could be a DTL-inverter, and if it is, the analysis has provided a causal explanation for how the circuit achieves that function. The circuit has been recognized to be a DTL-inverter.

Interpretations

The analysis process usually discovers multiple values for the circuit quantities. If two of these values differ and have compatible environments, a contradiction is recorded. (Note that a value has the three parts of IQ expression, environment and derivation, and that values are compared by the their IQ expressions.) In the case where one of the environments is a subset of the other, one or both of these values will immediately stop propagating. Although contradictions rule out most of the multiple values, many cells still contain multiple, possibly differing values at the conclusion of the analysis. If all of the values in a particular cell are the same, then no further analysis is necessary since the value holds independently of any environment. However, if the values differ, the correct environment need to be disambiguated in order to determine the correct value.

In order to identify the unique causal argument which describes how circuit output is related to circuit input, the environments have to be disambiguated regardless of whether all the output values are the same or not. More components may contribute to the behavior of the circuit than appear in the causal argument of the output. Since the purpose of these components must be identified, multiple values in these regions of the circuit must also be disambiguated. The disambiguation of every circuit quantity is summarized by an environment consisting of a maximal collection of assumptions which selects noncontradictory values from each cell. Such an environment is called an *interpretation*. An interpretation selects a value if the environment of that value is a subset of the interpretation. The maximality condition ensures that the addition of any assumption causes the interpretation to select contradictory values from some cell. In order to accommodate the behavior of those components whose local causal flow is indeterminate, the interpretation may contain incompatibilities.

At the conclusion of the analysis those cells which have not received values are presumed to contain zero. The rationale is that an effect must have a cause, and all possible causes have been explored. The connection heuristics make the implicit assumption that all unknown quantities are zero, so there is no necessity for propagating these values. An interpretation may select no value for a cell and thus take advantage of the fact that a cell with no values is zero. If a cell has no values under a certain interpretation, no cause has been found for it, and therefore it is presumed to be zero.

The process of causal analysis explores all possible interpretations of a circuit's behavior. Although it is good at determining causal arguments within a particular interpretation, it is bad at identifying which interpretation is the correct one. Some assumptions can be verified by causal reasoning and other assumptions critically depend on parameter values, but the verification of most assumptions requires fundamentally different analysis techniques. The latter are based on more complicated reasoning about constraints and purposes. One way to avoid applying these techniques is to build the circuit and take measurements.

Two of the arguments causal analysis finds for the output behavior of the feedback amplifier are:

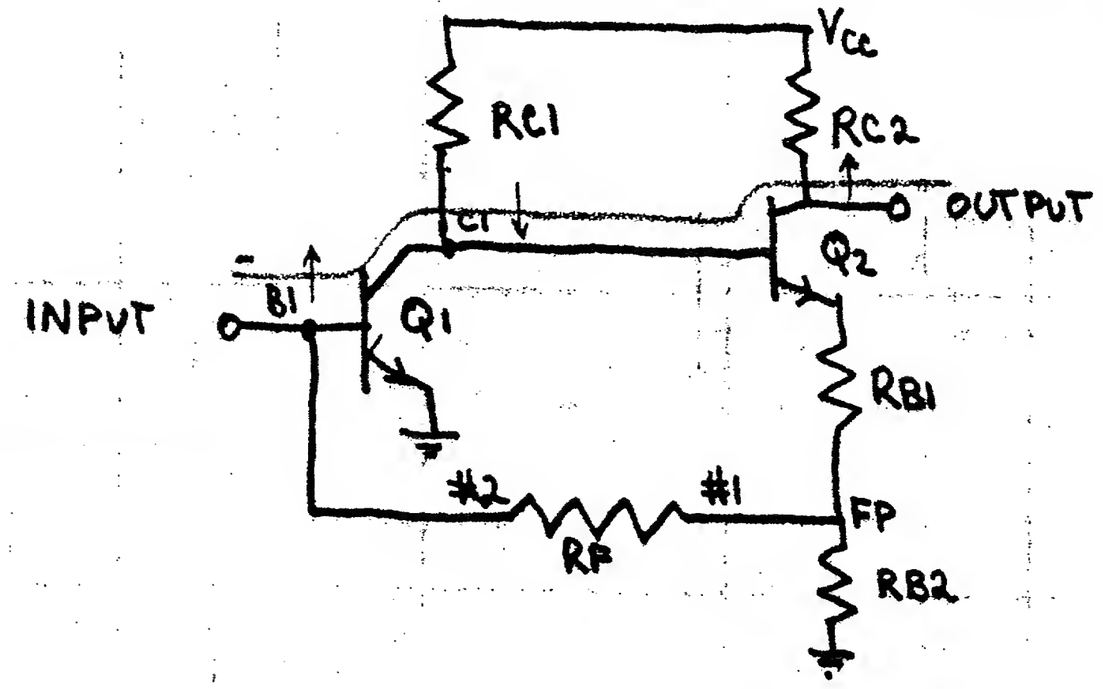

Figure 14 : Feedback Amplifier

```
Starting with input:
(CURRENT TERMINAL IN) = ↓  <>
   Premise.
(VOLTAGE B1 GROUND) = ↑  <[B1 IN]>
   KCL-heuristic [B1 IN]
(CURRENT C Q1) = ↑  <[B1 IN] (Q1 ON)>
   V => IC for Q1
(VOLTAGE C1 GROUND) = ↓  <[C1 Q1] [B1 IN] (Q1 ON)>
   KCL-heuristic [C1 Q1]
(VOLTAGE E2 C1) = ↑  <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   KVL-heuristic [Q2 VB]
(CURRENT C Q2) = ↓  <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   V => IC for Q2
(CURRENT #2 RC2) = ↑  <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   KCL for node OUTPUT
(CURRENT #1 RC2) = ↓  <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   KCL for device RC2
(VOLTAGE OUTPUT VCC) = ↑  <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   = V I for RC2
Also given that:
(VOLTAGE VCC GROUND) = 0  <>
   SUPPLY
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT VCC) cause:
(VOLTAGE OUTPUT GROUND) = ↑  <[Q2 VB] [-C1 Q1] [+B1 IN] (Q2 ON) (Q1 ON)>
   KVL applied to nodes OUTPUT VCC GROUND
```

The increased input voltage turns Q1 on harder, pulling down its collector. This falling voltage is applied to the base of Q2, causing it to begin to turn off. Since Q2's collector current is dropping, the voltage across the load RL must also drop.

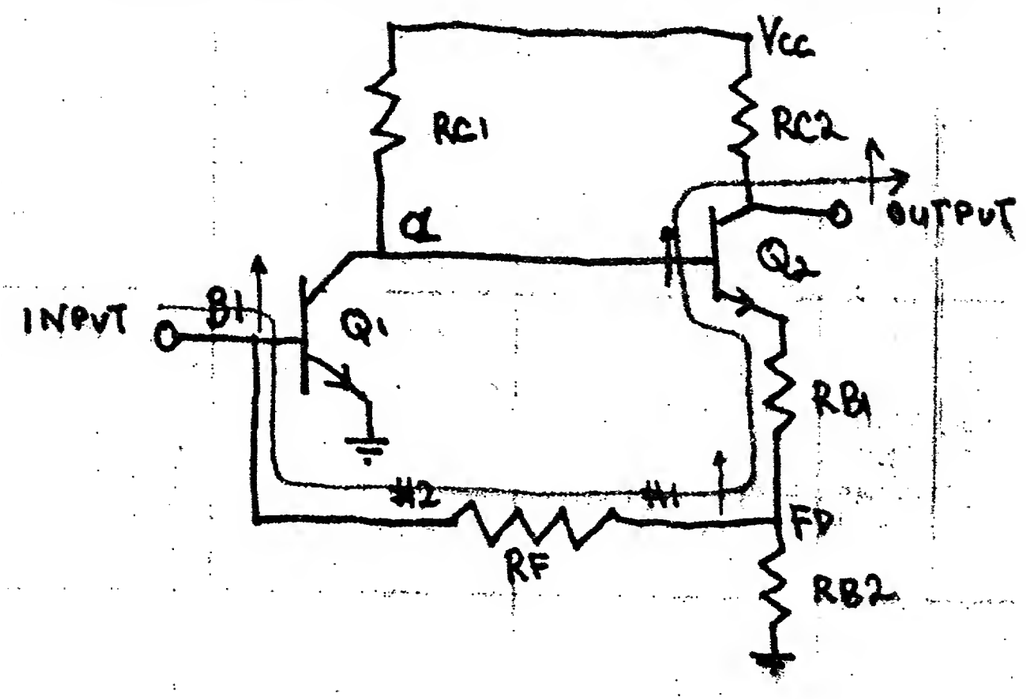

Figure 15 : Feedback Amplifier

Starting with input:

```
(CURRENT TERMINAL IN) = ↓  <>
   Premise.
(VOLTAGE B1 GROUND) = ↑  <[B1 IN]>
   KCL-heuristic [B1 IN]
(VOLTAGE FP B1) = ↓  <[RF V2] [B1 IN]>
   KVL-heuristic [RF V2]
(CURRENT #1 RF) = ↓  <[RF V2] [B1 IN]>
   V => I for RF
(VOLTAGE FP GROUND) = ↑  <[FP RF] [RF V2] [B1 IN]>
   KCL-heuristic [FP RF]
(VOLTAGE FP E2) = ↑  <[RB1 V2] [FP RF] [RF V2] [B1 IN]>
   KVL-heuristic [RB1 V2]
(CURRENT #1 RB1) = ↓  <[RB1 V2] [FP RF] [RF V2] [B1 IN]>
   V => I for RB1
(VOLTAGE E2 GROUND) = ↑  <[E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN]>
   KCL-heuristic [E2 RB1]
(VOLTAGE E2 C1) = ↑  <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KVL-heuristic [Q2 VE]
(CURRENT C Q2) = ↓  <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   V => IC for Q2
(CURRENT #2 RC2) = ↑  <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KCL for node OUTPUT
(CURRENT #1 RC2) = ↓  <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KCL for device RC2
(VOLTAGE OUTPUT VCC) = ↑
        <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   = V I for RC2
```

Also assuming that:

```
(VOLTAGE VCC GROUND) = 0  <>
   SUPPLY
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT VCC) cause:
(VOLTAGE OUTPUT GROUND) = ↑
        <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KVL applied to nodes OUTPUT VCC GROUND
```

The increased input voltage is coupled through RF and RB1 to the emitter of Q2. The rising voltage at the emitter causes Q2 to begin to turn off, consequently lowering its collector current. The voltage across the load RL must also drop.

The circuit's behavior has four interpretations:

```
<[B1 IN] [RB1 V1] [RF V2] [FP RF] [RB1 V2] [E2 Q2] [Q2 VB] [C1 Q1]>
<[B1 IN] [Q2 VE] [RF V2] [FP RF] [RB1 V2] [E2 RB1] [Q2 VB] [C1 Q1]>
<[B1 IN] [RF V2] [FP RF] [RB1 V2] [E2 RB1] [Q2 VE] [C1 Q2]>
<[B1 IN] [RF V1] [E2 Q2] [RF V2] [RB1 V1] [FP RB1] [Q2 VB] [C1 Q1]>
```

The four interpretations originate from the circled ambiguities:

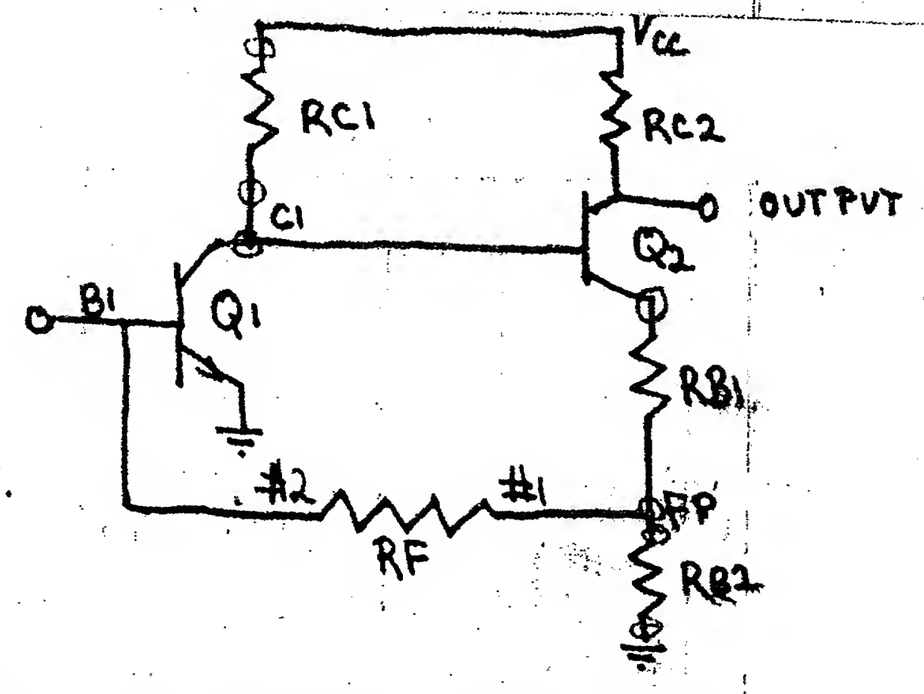

Figure 16 : Feedback Amplifier Ambiguities

An ambiguity is *minimal* if there is no other simpler ambiguity whose resolution would automatically resolve it as well. Two ambiguities are *similar* if they involve identical environments. If only the minimal instance of each ambiguity is retained, and if similar ambiguities are grouped together, only three ambiguities remain:

```
Ambiguity between
<[B1 IN] [E2 RB1] [RB1 V2] [FP RF] [RF V2]> resolves 2 other ambiguities
<[B1 IN] [E2 Q2] [Q2 VB] (Q2 ON) [C1 Q1] (Q1 ON)> resolves 1 other ambiguities
occurs at
(VOLTAGE E2 GROUND)
(VOLTAGE E2 VCC)
```

```
Ambiguity between
<[B1 IN] [FP RF] [RF V2]> resolves 3 other ambiguities
<[B1 IN] [FP RB1] [Q2 VB] (Q2 ON) [C1 Q1] (Q1 ON)> resolves 1 other ambiguities
occurs at
(VOLTAGE FP GROUND)
(VOLTAGE FP VCC)
(CURRENT #1 RB2)
(CURRENT #2 RB2)

Ambiguity between
<[B1 IN] [C1 Q1] (Q1 ON)> resolves 3 other ambiguities
<[B1 IN] [C1 Q2] [Q2 VE] (Q2 ON) [E2 RB1] [RB1 V2] [FP RF] [RF V2]>
        resolves 1 other ambiguities
occurs at
(VOLTAGE C1 GROUND)
(VOLTAGE C1 VCC)
(CURRENT #1 RC1)
(CURRENT #2 RC1)
```

Through careful analysis of the environments, the number of measurements required to resolve the ambiguities can be minimized. Each measurement will contradict one of the two environments of an ambiguity. The contradiction of any particular environment may also automatically resolve other ambiguities. For example, if environment <[C1 Q1] (Q1 ON)> of the third ambiguity is contradicted, all the other ambiguities are automatically resolved since each other ambiguity has one environment which contains <[C1 Q1] (Q1 ON)>. This number is indicated after the environment in the summary. Since there is no *a priori* information about which environments will be contradicted, the next ambiguity to resolve is selected on the basis of the average number of ambiguities that would be resolved by the measurement. By this measure the second and third ambiguities have better scores, and QUAL arbitrarily picks the second. Voltage measurements are usually easier to take, so QUAL asks for one of the two voltages that would resolve the ambiguity:

```
Optimal voltage measurements are:
(VOLTAGE FP VCC)
(VOLTAGE C1 VCC)
Is the value of (VOLTAGE FP VCC) ↑ or ↓
```

The voltage at FP is observed to be falling. The final consistent interpretation for the circuit's behavior is:

<[B1 IN] [RF V1] [E2 Q2] [RF V2] [RB1 V1] [FP RB1] [Q2 VB] [C1 Q1]>

Since all the possible assumptions about RF are included in this interpretation, the causality around RF has not been clarified.

This interpretation identifies the correct causal argument for the output presented in Figure 14. Since causal analysis did not determine the correct interpretation, the selected causal

argument remains a rationalization of the observed behavior. Causal analysis assigns multiple values to circuit quantities only if they can be derived in multiple ways. This only happens if the circuit contains possible feedback paths. Since ambiguities always stem from possible feedback paths, explicit knowledge about feedback should be incorporated into the analysis process.

Fault Localization

This paper has described the beginnings of a theory of what it means to understand how a circuit works. One test of such a theory must be whether this understanding can be utilized to analyze circuit faults. One use of fault localization techniques is *troubleshooting*. Troubleshooting involves determining why a particular correctly designed circuit is not functioning as intended, the explanation for the faulty behavior being that the particular instance of that circuit under consideration is at variance in some way with its design. The same techniques are also applicable to *debugging* almost correct designs [Sussman 77]. If the designer has a description of how the circuit should behave and has an implementation of that behavior that is correct except for some small local problem, the intentions of the designer can be used to determine which component is contributing to the unintended behavior. The success of the fault localization strategies discussed here will depend on having a description of how the circuit should work and on whether the fault is localized to a small area of the circuit.

Every mechanism which can predict behavior can be utilized to predict the new behavior which would result if a fault were introduced. Troubleshooting by synthesis exhaustively hypothesizes all possible faults and eliminates those faults which are not consistent with the observed symptoms. This technique is computationally impractical with conventional circuit analysis programs. Moreover, special techniques have to be developed to cope with the infinite number of faults a single component can have (e.g. a resistor can have an infinite number of incorrect values). Since causal analysis uses a simple algebra, the computation is more tractable and the number of faults a component can have is limited.

There are two different techniques for evaluating hypothetical faults. A faulty model can be used in the usual causal analysis to determine whether the predicted behavior is consistent with the observed symptoms. This technique would determine that a failing DTL-inverter could be explained by D1 being stuck off. Another technique is to remove the input signal and treat the inserted fault as the signal. This technique predicts the change in quiescent behavior. If the predicted change is consistent with the difference between the correct and observed quiescent behavior, the fault explains the symptoms. For example, suppose the beta of the DTL-inverter output transistor is too low. Introducing this fault in the state when all the devices are on, the collector current decrements. This explains the quiescent fault that the inverter is not pulling down hard enough. The latter technique is particularly useful in identifying faults in the quiescent aspects of the circuit behavior, and the technique is useful for identifying faults in the incremental behavior. Unfortunately, neither technique provides a method making hypotheses. Troubleshooting by synthesis using these evaluation techniques is inefficient both in terms of computational resources and in the number of measurements required to isolate the faulted component.

Since causal analysis usually finds multiple interpretations for the behavior, these two techniques work considerably better when the correct interpretation is known. The interpretation can be used to indicate which states to examine for symptomatic behavior. For example, D1 stuck off explains the circuit's inability to invert only if inversion takes place in the state in which all the devices are on. The more that is known about the circuit's behavior, the easier it is to troubleshoot it. The expected input-output behavior is necessary to determine that a fault exists at all, and knowledge of the correct interpretation guides the analysis of hypothetical faults. By making random measurements, troubleshooting by synthesis will eventually localize the fault, but it is more profitably used as an hypothesis evaluator for the localization strategies.

The interpretation also provides a causal explanation for how the outputs are caused by the inputs. The devices mentioned in this explanation are prime candidates for possible faults and the fault modes can be determined by examining the argument. The resulting hypotheses can be evaluated to determine which faults in which of these devices are consistent with the symptoms. The difficulty with this is that the interpretation may be changed by the presence of the fault. If a causal assumption is violated, the entire argument may be invalidated because the dominant effect may actually be the quantity which caused the violation. Since the designer never intended that the circuit behave in that way, no appeal can be made to the original intention. Similarly, state diagram heuristics which apply to working circuits cannot be used. The interpretation under which the faulty circuit is behaving must be disambiguated by actual measurements. The procedure presented in the previous section can determine the interpretation by taking appropriate measurements. When the new interpretation is identified, its causal argument can be examined for faults.

If the behavior prediction mechanism is invertible, this property can be utilized for fault localization and for design; the symptomatic or desired input-output behavior is used as an input to the inverted prediction mechanism in order to identify faults in or constraints on the individual components. Numerical techniques are not invertible and therefore inapplicable. Propagation of symbolic constraints can be quite successful in synthesizing a circuit from a desired input-output behavior [de Kleer & Sussman 78], but it is not as applicable to troubleshooting [de Kleer 75]. When desired behavior differs from expected behavior, blame can be assigned to any device involved in the propagation. In a detailed analysis the desired output behavior may depend on every circuit device, therefore the observed symptom provides no information. The strategy only becomes informative after internal measurements have been taken which introduce sufficient redundancy that the constraints do not need to depend upon every device in the circuit. Even after some internal measurements have been taken, the strategy is incapable of suggesting further measurements to take. Some other mechanism must be employed to suggest informative measurements.

A particular causal argument can be inverted to determine what could have caused the undesirable output. However, far more profit can be made by inverting the causal analysis process itself. The direction of time flow can be reversed in the analysis process in order to determine what could have caused the undesirable behavior. The direction of time flow is primarily provided by the models, and these can be easily inverted. For example, an increased transistor v_{BE} is used to

derive an increased i_C but not vice versa. When the direction of time flow is reversed, i_C is used to derive an increased v_{BE} but not vice versa. In forward time a deduction "A implies B" signifies "A causes B" while in reverse time it signifies "A can be caused by B." The inverted model for a transistor is:

```
(choice (on ()
            (↕ic (=> ↕v))
            (↕ie (-=> ↕v))
            (↕ib (=> ↕v)))
        (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie)))
        (sat ((0 => ↕ic))))
```

The other device models are easily inverted.

KCL and KVL remain unchanged. The connection heuristics require major modification. The KVL-heuristic is easily dealt with. In forward time analysis a device model can be triggered by a voltage-to-reference on one of its input nodes. The reverse time KVL-heuristic deduces a voltage-to-reference whenever the inverted device model determines a voltage on an input. For example, the forward KVL-heuristic triggers the transistor rule on the assumption that the base voltage is dominant, and the reverse KVL-heuristic deduces the voltage on the base from the collector current under the assumption that the base voltage was the dominant input that caused the collector current. The assumption is recorded as $[Q\ v_B]$ in both cases. In order to understand the reverse KCL-heuristic reconsider the network theory behind the assumption:

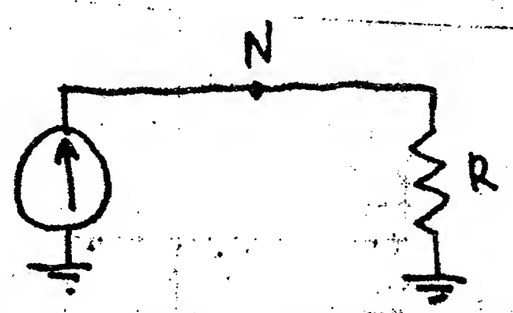

Figure 17 : KCL-heuristic Network Assumption

The forward KCL-heuristic makes the assumption that the unknown currents into a node behave as a positive resistance so that it can predict the voltage at that node. The reverse KCL-heuristic is derived by applying Ohm's law to this positive resistance. In reverse analysis, when a voltage at a node is discovered, it is assumed to be the result of current flowing through the positive resistance. The reverse KCL-heuristic is implemented by a procedure which is triggered whenever a voltage-to-reference is discovered and then assumes that the unknown terminal currents individually receive the current which Ohm's law predicts the entire unknown bundle of terminals should receive. The reverse KCL-heuristic is as complicated as the forward KCL-heuristic because it must carefully analyze the environments of the voltage and the currents.

When time is reversed, the ambiguities in the analysis result from the inability to identify the specific cause for an effect. These ambiguities can be handled by the same methods used in forward time reasoning to handle causes with uncertain effects. Compare the following reverse time causal analysis of the DTL-inverter with the earlier forward time causal argument.

Starting with input:

```
(VOLTAGE OUTPUT GROUND) = ↓  <>
     Premise.
Also given that:
   (VOLTAGE GROUND VCC) = 0  <>
     POSITIVE-SUPPLY
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT GROUND) cause:
   (VOLTAGE OUTPUT VCC) = ↓  <>
   KVL applied to nodes GROUND VCC OUTPUT
   (CURRENT #1 RL) = ↑  <>
     = V I for RL
   (CURRENT #2 RL) = ↓  <>
     KCL for device RL
   (CURRENT C Q1) = ↑  <>
     KCL for node OUTPUT
   (VOLTAGE B GROUND) = ↑ <(Q1 ON)>
     IC => V for Q1
   (CURRENT #2 D2) = ↓  <[B D2] (Q1 ON)>
     KCL-heuristic [B D2]
   (CURRENT #1 D2) = ↑  <[B D2] (Q1 ON)>
     KCL for device D2
   (VOLTAGE B N) = ↓  <[B D2] (Q1 ON)>
     I => V for D2
   (VOLTAGE N GROUND) = ↑  <(D2 ON) [D2 V1] [B D2] (Q1 ON)>
     KVL-heuristic [D2. V1]
   (CURRENT #1 D1) = ↓  <[N D1] (D2 ON) [D2 V1] [B D2] (Q1 ON)>
     KCL-heuristic [N D1]
   (VOLTAGE N INPUT) = ↓  <(D1 ON) [N D1] (D2 ON) [D2 V1] [B D2] (Q1 ON)>
     V => I for D1
   (VOLTAGE INPUT GROUND) = ↑
     <[D1 V2] (D1 ON) [N D1] (D2 ON) [D2 V1] [B D2] (Q1 ON)>
   KVL-heuristic [D1 V2]
```

This explanation lists the events in the usual order of discovery that was used for the forward time explanations. Event A is an antecedent of event B if B takes part in a possible causal deduction of A. Note that the above explanation is just the inverse of the forward time explanation.

The forward time flow analysis is an information-losing process; any value which is not an output and does not propagate is lost. Contradictions have no direct effect on the output signal and are also lost. Since the reverse time analysis is given only one piece of information about the forward time flow behavior, it cannot analyze the entire circuit. However, it should be able to find a causal argument to explain what inputs could have caused observed outputs. It can use forward time flow analysis to check the interpretation it discovers.

In the reverse time analysis, the faulty output can be caused by either faulty inputs to the component or the component itself. Just as in troubleshooting by synthesis, there are two different techniques for using the strategy. The undesirable difference between observed and expected quiescent behavior can be treated as the quantity to be explained. Using this technique, a low collector current is explained by a low beta. The other technique explains the undesired response to the applied signal directly, explaining a positive gain in the inverter by a possible base-collector short in the output transistor. Since most faults manifest themselves quiescently, the first technique is generally more useful.

Except for assumptions at the external connections the interpretation for a behavior is independent of the direction of time flow. Nevertheless, a fault may force a different unintended interpretation. In order to disambiguate the interpretations, measurements internal to the circuit must be taken. Applying this strategy the fault localization process takes circuit measurements for two purposes. When a possible causal explanation for the symptomatic behavior is known, measurements are necessary to determine which device in this explanation could be faulted. If no interpretation is known, or if measurements invalidate an interpretation, measurements must be taken to determine a new interpretation.

The reverse time localization process is considerably different than troubleshooting by synthesis. It makes only one analysis with the undesired behavior as the input signal, while troubleshooting by synthesis has to do a separate analysis for every possible fault. Although the causal argument they both eventually arrive at to explain the symptomatic behavior is isomorphic (under time-reversal), the reverse time strategy has made a more efficient set of measurements and is able to explain why the measurements were made and why other faults were not considered. The only explanation troubleshooting by synthesis can provide to explain why a device is not faulted is that a fault in the device is not consistent with the observed symptoms. It is also poor at suggesting further measurements.

Both localization strategies are successful, and their success is due in large part to utilizing the knowledge of how the circuit works. The missing piece of the theory is hierarchy. The strategies discussed in this section apply to any level of detail, but they do not explain how to move between levels of detail. An improved localization system would first analyze the fault at the shallowest level of detail. After the fault has been localized to particular modules, it would consider the implementation of only those models which could contain the fault.

The Relationship Between Causality and Constraint

There is an interleaved hierarchy of causal and constraint-like descriptions for the same physical phenomena. In the Schmitt trigger description the engineer uses a causal description. In order to determine the precise values of the electrical quantities he will employ a constraint representation consisting of algebraic equations. This lumped-parameter representation is modeled on more basic causal phenomena. Although constraint-like representations have been explored to a great extent, little attention has been paid to the more causal representations that people prefer.

In contrast to causal arguments, a quantitative description of a system's behavior is in terms of a set of quasistatic constraints describing the dynamics of the system. Indeed, the lumped-

parameter circuit model of the physical system described by a circuit diagram is only valid under the assumption that the system is always at equilibrium. But a circuit is only useful because its equilibrium changes under the influence of imposed signals. The "force" that moves a circuit from one interesting equilibrium to another, when driven by a signal, is that the incremental signal slightly displaces the equilibrium from the circuit's state. The process of equilibrating is adequately described by the differential equations of the dynamics of the circuit. The manner in which the signal moves the equilibrium around is better described by the qualitative, causal arguments.

Although the equilibrating process can be quantitatively described, it is difficult to quantitatively describe the manner in which the signal moves the equilibrium. The lumped-parameter circuit model is an idealization and simplification of the behavior of the electromagnetic fields in and around the circuit components. Since changes in these fields propagate at finite speeds, this process takes a certain amount of time. The differential equations of the lumped-parameter circuit model cannot account for what happens during this period of disequilibrium. Within this period the changing fields of the input signals propagate until global equilibrium is reached. This propagation can be viewed as a kind of causal flow: the input field changes and propagates to other materials causing further fields to change. These changes can be partially ordered in a time sequence in which each change is caused by changes earlier in the sequence and earlier in time.

The quantitative calculation of the causal flow that happens during the period of disequilibrium is intractable. Although the electromagnetic laws that govern the physics are known, there is no practical way to quantitatively describe this causal process. This means that the electrical engineer can never completely analyze a circuit. Fortunately the engineer is only interested in analyzing the circuit to a certain amount of precision and the lumped-circuit model provides a technique for this. If the disequilibrium in a particular area of the circuit is important to the overall behavior, the engineer introduces *parasitic* capacitors and inductors to describe this disequilibrium. This technique captures the quasistatic effects of the disequilibrium but not the causal effects.

Although causal reasoning is quantitatively intractable, it appears to be the preferred mode of reasoning for humans. Feynman [65] makes a similar observation about how people prefer to think about the Law of Gravitation. When an electrical engineer reasons about a circuit he tries to reintroduce the causality that the lumped-circuit model throws away. He does this by using locally causal models and imposing a time flow on the changes in circuit quantities. Only by throwing away most of the detail of the models and the causality is he able to make the causal analysis tractable. The engineers' qualitative theory of circuit causality explains the period of equilibration by introducing finite time flow and permitting the circuit to be in disequilibrium. The actual lumped-circuit model of the circuit he uses allows him to include only those effects that are important, and his causal argument describes the effect of each component in the disequilibrium period.

Topology and Geometry

This research focuses on analyzing a circuit by simulating its behavior. This is called

functional analysis. A circuit can also be analyzed by topological and geometric techniques. *Topological analysis* compares the topology of the circuit with previously recognized topologies. *Geometric analysis* relies on the tacit graphical language engineers use when they describe circuit topologies on paper. This research focuses on functional analysis because topological and geometric techniques break down when presented with circuits sufficiently different from those previously recognized. Since the ultimate purpose of the analysis is to explain behavior, topological analysis must translate its results into behavioral terms. This means there must be a way to specify the input-output behavior of a fragment of topology. If the internal behavior of a recognized circuit fragment is to contribute to the causal explanation of the circuit's behavior, the fragment must have been functionally analyzed at one time.

Topological recognition must, of course, play a role in recognizing the components themselves. There are also many local topological transformations which are useful. For example, a real DTL inverter employs a series of diodes for D2. In the IQ model, two diodes in series or in parallel function as one diode. The following is a collection of some of the more common local topological transformations:

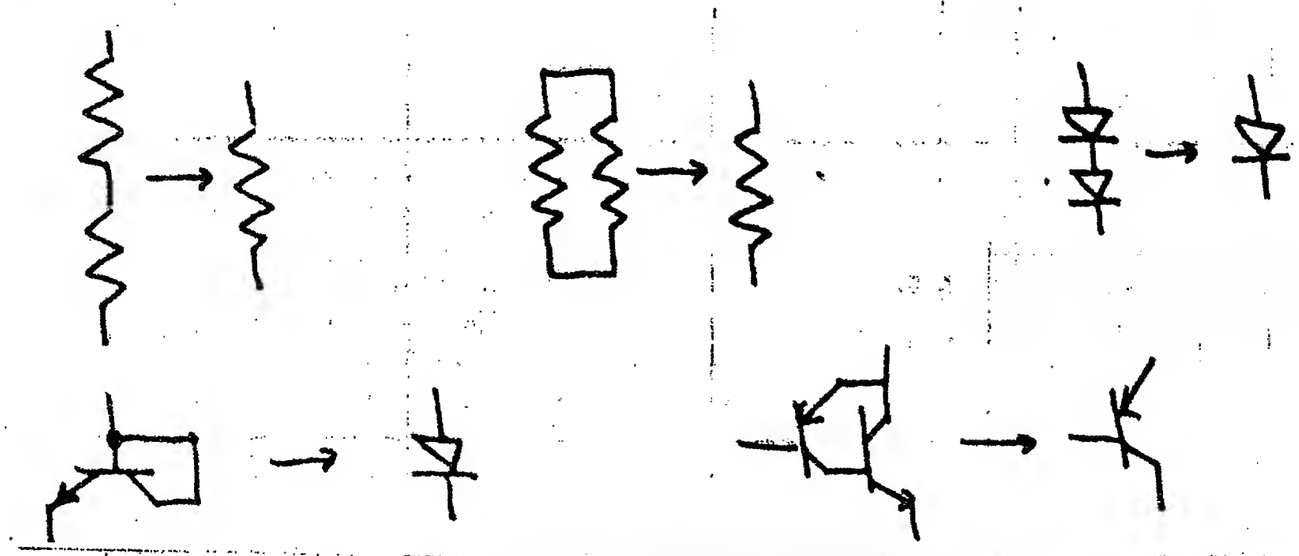

Figure 18 : Local Topological Transformations

The technique of substituting one topology for another becomes much less useful when applied to larger fragments since the same large fragment will rarely be seen again. The next instance of the fragment will probably fail to match. To be useful, topological analysis must be able to the make the partial matches. For example, the following three circuits are all instances of emitter-coupled pairs:

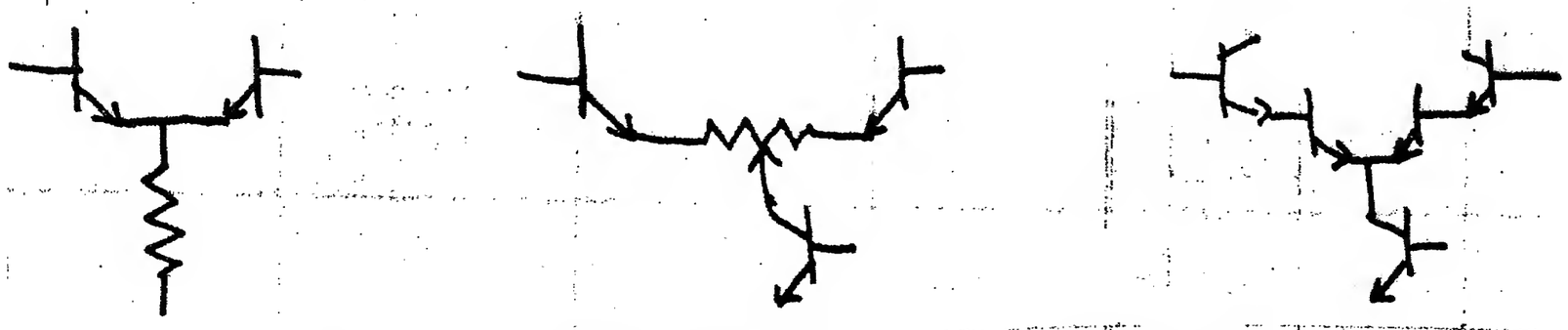

Figure 19 : Emitter-Coupled Pairs

Current partial match strategies are rather poor. In order to be general the matcher must

incorporate functional information. There are also only a small number of fragments whose behavior can be summarized with simpler topologies.

Engineers make extensive use of topological and geometric analysis in recognizing new circuits. The engineer may recognize some fragments topologically and use functional analysis to identify others. Often a fragment, or perhaps an entire circuit, is so similar to a previously recognized fragment that the old fragment's behavioral description is used, augmented by the impact of the topological differences. In short, the engineer's recognition process depends equally on topological and functional analysis. Nevertheless, the final explanation is always a behavioral one which often does not refer to the recognized fragments, but only includes that aspect of their internal behavior which is important.

The geometry of the circuit schematic has tremendous impact on the engineer's topological recognition process. Electrical engineering has a tacit language of expressing topologies that is rigidly followed. Circuits of a particular type are always drawn in the same way. Geometric analysis can also guide the functional analysis. Causal analysis often gets confused about the direction of causality in feedback paths. The main signal path usually has more components along it. The feedback path usually involves a long line with few components attached to it. Signals usually flow from left to right, and feedback paths usually flow from right to left.

Applications

The causal analysis process presented in this paper is capable of analyzing most circuits. The arguments QUAL discovers to explain circuit behavior are similar to those engineers find. Causal analysis is also shown to be useful for recognition and troubleshooting. There are a number of other directions in which QUAL could be naturally extended. QUAL does not currently deal with true time delays introduced by capacitors and inductors; an understanding of capacitors and inductors is required to understand oscillators. QUAL can be used to determine the appropriate models and places to introduce variables for algebraic analysis. By specifying constraints on the causal behavior, QUAL can determine algebraic constraints on the circuit parameters. A qualitative quiescent analysis would be very useful for identifying contradictory states.

QUAL is based on a far from complete theory of understanding circuit behavior. QUAL is capable of simulating the behavior of a circuit at a deep enough level to be useful, and a shallow enough level to be computationally tractable. This analysis by simulation is very useful, but analysis has a more general purpose. The behavior of the circuit is the manifestation of a particular plan which is implemented by the circuit, and the analysis of a circuit involves discovering this plan. The engineer brings a library of such plans to bear when he encounters new circuits. Analysis requires using this library to identify the plan which explains the circuit's behavior.

Analysis must determine the behavior of the circuit and purposes of the components. Behavioral teleology explains the purpose of a component according to how it contributes to the causal argument which explains the circuit's behavior. Implementation teleology relates the particular circuit to its plan. The interpretation provides a mechanism for representing behavioral

teleology. The causal argument indicated by a particular interpretation specifies how each component contributes to the circuit's input-output behavior. The interpretations for the different circuit states is a starting point for identifying the plan for the circuit's behavior.

For a more detailed discussion of a possible theory of plans for circuits see [de Kleer 77]. The theory of plans discussed there is motivated by the theory of plans being developed for programs [Rich *et.al.* 77]. Rieger and Grinberg [77] have developed a method for explicitly representing the causality of physical mechanisms. The theory of plans being developed here draws on both of these, employing the notions of teleology developed in programming, while preserving the causality of the basic mechanisms.

The most important class of plans involves feedback. Feedback is the basic mechanism by which the behavior of a collection of components can be controlled. The inherently global nature of feedback makes it difficult to deal with directly using causal analysis. However, the causal explanations of the propagations can be examined to detect feedback. Feedback occurs if a cell gets a value which depends on a previous value of the cell. This can only happen for a causal assumption made by a connection heuristic. For example, QUAL may propagate a signal along a main signal path assuming the feedback connection is nondominant. When the main signal propagates a value back along the feedback path, this assumption will be violated and feedback is detected. The detection of feedback is possible as a result of the connection heuristics.

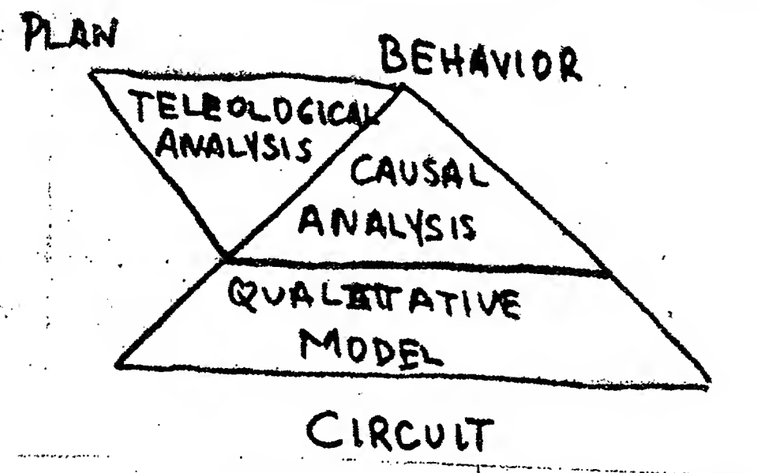

Figure 20 : Analysis

Causality and constraint play key roles in the understanding of the physical world. At some level of detail theories are either completely causal or completely constraint-like. At the qualitative level at which people reason about the world, some domains are more causal than others. Constraint plays a minor role in the roller coaster world of NEWTON [de Kleer 75]. NEWTON uses a kind of causality to envision different possible successor scenes and then uses constraint analysis to determine which scene is correct. It is largely an artifact of the the roller coaster domain that the constraints are only used algebraically.

The applicability of the theory of causal reasoning presented in this paper depends on what extent to which the domain exhibits the following two properties.

(1) All the potential cause-effect interactions must be determinable *a priori*, and the number of such interactions must be finite. In other words, the topology of all the possible causal interactions must be determined before causal analysis starts.

(2) The domain has constraint-like and causal inferences.

The first property is violated in the problems of a pendulum swinging against a nail, a truck wedged in a tunnel, or a ball bouncing inside of a cube. In all of these problems the precise point of causal interaction cannot be easily determined and it is impossible to *a priori* describe a topology of causal interactions. The extent to which the second property holds, determines the richness of the interaction between causal and constraint-like reasoning. Electronics is an optimum domain in which to explore causal reasoning since it admits a rich constraint-like and causal analysis. As a more precise theory of causal reasoning is developed for circuits, more insight will be obtained into the role causal reasoning plays in more complex reasoning about the causality of physical objects.

References

[Brown 76]
Brown, A.L., "Qualitative Knowledge, Causal Reasoning, and the Localization of Failures", Artificial Intelligence Laboratory, TR-362, Cambridge: M.I.T., 1976.

[Brown & Sussman 74]
Brown, A.L. and G.J. Sussman, "Localization of Failures in Radio Circuits -- A Study in Causal and Teleological Reasoning", Artificial Intelligence Laboratory, AIM-319, Cambridge: M.I.T., 1974.

[Brown *et.al.* 74]
Brown, J.S., R.R. Burton and A. G. Bell, *SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An example of AI in CAI)*, Final Report, B.B.N. Report 279, A.I. Report 12, Cambridge: B.B.N., 1974.

[Brown *et.al.* 76]
Brown, J.S., R. Rubinstein and R. Burton, *Reactive Learning Environment for Computer Assisted Electronics Instruction*, B.B.N. Report 3314, I.C.A.I. Report 1, Cambridge: B.B.N., 1976.

[de Kleer 75]
de Kleer, J., "Qualitative and Quantitative Knowledge in Classical Mechanics", Artificial Intelligence Laboratory, TR-352, Cambridge: M.I.T., 1975.

[de Kleer 76]
de Kleer, J., "Local Methods for Localizing Faults in Electronic Circuits", Artificial Intelligence Laboratory, AIM-394, Cambridge: M.I.T., 1976.

[de Kleer 77]
de Kleer, J., "A Theory of Plans for Electronic Circuits", Artificial Intelligence Laboratory, WP-144, Cambridge: M.I.T., 1977.

[de Kleer & Sussman 78]
de Kleer, J. and G.J. Sussman, "Propagation of Constraints Applied to Circuit Synthesis", Artificial Intelligence Laboratory, AIM-485, Cambridge: M.I.T., 1978.

[Feynman 65]
Feynman, R., *The Character of Physical Law*, Cambridge: M.I.T. Press, 1965.

[Harris *et.al.* 66]
Harris, J.N., P.E. Gray and C.L. Searle, *Digital Transistor Circuits*, New York: John Wiley & Sons, 1966.

[McDermott 76]
McDermott, D.V., "Flexibility and Efficiency in a Computer Program for Designing Circuits", Artificial Intelligence Laboratory, TR-402, Cambridge: M.I.T., 1976.

[Rich *et.al.* 77]
Rich, C., Shrobe H.E., Waters R.C., Sussman G.J. and Hewitt C.E., "Programming Viewed as an Engineering Activity", Artificial Intelligence Laboratory, AIM-459, Cambridge: M.I.T., 1977.

[Rieger & Grinberg 77]
Rieger, C., and M. Grinberg, "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms", *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, sp. 250-255, 1977.

[Stallman & Sussman 76]
Stallman, R.M. and G.J. Sussman, "Forward Reasoning and Dependency-Directed Backtracking In a System for Computer-Aided Circuit Analysis", Artificial Intelligence Laboratory, AIM-380, Cambridge: M.I.T., 1976.

[Sussman 77]
Sussman, G.J., "Electrical Design: A Problem for Artificial Intelligence Research", Artificial Intelligence Laboratory, AIM-425, Cambridge: M.I.T., 1977.